

# Cookies on GOV.UK One Login

We use some essential cookies to make this service work.

We'd like to set additional cookies so we can remember your settings, understand how people use the service and make improvements.

[Accept additional cookies](#)

[Reject additional cookies](#)

[View cookies \(<https://signin.account.gov.uk/cookies>\)](#)



**BETA** This is a new service. Help us improve it and [give your feedback \(opens in a new tab\) \(\[https://signin.account.gov.uk/contact-us-questions?theme=suggestions\\\_feedback\]\(https://signin.account.gov.uk/contact-us-questions?theme=suggestions\_feedback\)\)](#).

English | [Cymraeg \(/services-using-one-login?lng=cy\)](#)

## Services you can use with GOV.UK One Login

At the moment, you can only use GOV.UK One Login to access some government services.

In the future, you'll be able to use it to access all services on GOV.UK.

### Search for a service

Sorted by A to Z

49 results

[Apply for a Gender Recognition Certificate \(<https://www.gov.uk/apply-gender-recognition-certificate/how-to-apply>\)](#)

[Apply for a vehicle operator licence \(<https://www.gov.uk/apply-vehicle-operator-licence>\)](#)

[Apply for an export certificate  
\(https://www.gov.uk/guidance/apply-for-an-export-certificate\)](https://www.gov.uk/guidance/apply-for-an-export-certificate)

---

[Apply for an HM Armed Forces Veteran Card  
\(https://www.gov.uk/veteran-card\)](https://www.gov.uk/veteran-card)

---

[Apply for an import licence  
\(https://www.gov.uk/guidance/import-controls\)](https://www.gov.uk/guidance/import-controls)

---

[Apply for qualified teacher status in England \(https://apply-for-qts-in-england.education.gov.uk/eligibility/start\)](https://apply-for-qts-in-england.education.gov.uk/eligibility/start)

---

[Apply for teacher training in England  
\(https://www.gov.uk/apply-for-teacher-training\)](https://www.gov.uk/apply-for-teacher-training)

---

[Apply to become a registered social worker in England  
\(https://www.socialworkengland.org.uk/registration/apply-for-registration/\)](https://www.socialworkengland.org.uk/registration/apply-for-registration/)

---

[Apprenticeship assessment service  
\(https://www.gov.uk/guidance/apply-to-the-apar-as-an-apprenticeship-training-provider\)](https://www.gov.uk/guidance/apply-to-the-apar-as-an-apprenticeship-training-provider)

---

[Apprenticeship provider and assessment register \(APAR\)  
\(https://www.gov.uk/guidance/apply-to-the-apar-as-an-apprenticeship-training-provider\)](https://www.gov.uk/guidance/apply-to-the-apar-as-an-apprenticeship-training-provider)

---

[Cancel a lost or stolen passport \(https://www.gov.uk/report-a-lost-or-stolen-passport\)](https://www.gov.uk/report-a-lost-or-stolen-passport)

---

[Childcare Offer for Wales: parents and guardians  
\(https://www.gov.wales/get-30-hours-childcare-3-and-4-year-olds/apply\)](https://www.gov.wales/get-30-hours-childcare-3-and-4-year-olds/apply)

---

[Childcare Offer for Wales: providers  
\(https://www.gov.wales/register-your-childcare-setting-get-childcare-offer-wales\)](https://www.gov.wales/register-your-childcare-setting-get-childcare-offer-wales)

---

[Claim compensation if you were the victim of a violent crime \(https://www.gov.uk/claim-compensation-criminal-injury/make-claim\)](https://www.gov.uk/claim-compensation-criminal-injury/make-claim)

---

[Confirm my apprenticeship details  
\(https://my.apprenticeships.education.gov.uk/\)](https://my.apprenticeships.education.gov.uk/)

---

[Connectivity Tool from the Department for Transport  
\(https://www.gov.uk/guidance/connectivity-tool\)](https://www.gov.uk/guidance/connectivity-tool)

---

[Driver and vehicles account \(https://www.gov.uk/driver-  
vehicles-account\)](https://www.gov.uk/driver-vehicles-account)

---

[Driving with a medical condition  
\(https://www.gov.uk/browse/driving/disability-health-condition\)](https://www.gov.uk/browse/driving/disability-health-condition)

---

[Early years child development training \(https://child-  
development-training.education.gov.uk/\)](https://child-development-training.education.gov.uk/)

---

[Early years financial incentives  
\(https://www.gov.uk/government/collections/additional-payments-  
for-teaching-eligibility-and-payment-details\)](https://www.gov.uk/government/collections/additional-payments-for-teaching-eligibility-and-payment-details)

---

[Energy Savings Opportunity Scheme reporting  
\(https://manage-energy-saving-opportunities-  
reporting.service.gov.uk/\)](https://manage-energy-saving-opportunities-reporting.service.gov.uk/)

---

[Find a job in teaching or education in England  
\(https://www.gov.uk/find-teaching-job\)](https://www.gov.uk/find-teaching-job)

---

[Find a UK market conformity assessment body \(https://find-  
a-conformity-assessment-body.service.gov.uk/\)](https://find-a-conformity-assessment-body.service.gov.uk/)

---

[Find an apprenticeship in England \(https://www.gov.uk/apply-  
apprenticeship\)](https://www.gov.uk/apply-apprenticeship)

---

[Find and apply for a grant \(https://www.gov.uk/guidance/find-  
government-grants\)](https://www.gov.uk/guidance/find-government-grants)

---

[Find and update company information \(https://find-and-  
update.company-information.service.gov.uk/\)](https://find-and-update.company-information.service.gov.uk/)

---

[Find and use an API from the Department for Education  
\(https://beta-find-and-use-an-api.education.gov.uk/\)](https://beta-find-and-use-an-api.education.gov.uk/)

---

[Find high value contracts in the public sector \(Find a](#)

[Tender](https://www.gov.uk/find-tender) (<https://www.gov.uk/find-tender>)

---

[Find teacher training courses in England](https://www.gov.uk/find-teacher-training-courses)  
(<https://www.gov.uk/find-teacher-training-courses>)

---

[Foreign Influence Registration Scheme \(FIRS\)](https://www.gov.uk/government/collections/foreign-influence-registration-scheme)  
(<https://www.gov.uk/government/collections/foreign-influence-registration-scheme>)

---

[GOV.UK email subscriptions](https://www.gov.uk/email/manage/authenticate)  
(<https://www.gov.uk/email/manage/authenticate>)

---

[Manage apprenticeships](https://www.gov.uk/sign-in-apprenticeship-service-account) (<https://www.gov.uk/sign-in-apprenticeship-service-account>)

---

[Manage fishing permits and catch returns in Wales](https://www.gov.wales/apply-fishing-permit)  
(<https://www.gov.wales/apply-fishing-permit>)

---

[Manage your State Pension account](https://www.manage-state-pension.service.gov.uk/start) (<https://www.manage-state-pension.service.gov.uk/start>)

---

[Modern slavery statement registry](https://www.gov.uk/guidance/add-your-modern-slavery-statement-to-the-statement-registry)  
(<https://www.gov.uk/guidance/add-your-modern-slavery-statement-to-the-statement-registry>)

---

[Ofqual subject matter specialist account](https://www.gov.uk/guidance/subject-matter-specialists-for-ofqual)  
(<https://www.gov.uk/guidance/subject-matter-specialists-for-ofqual>)

---

[Product Safety Database](https://www.product-safety-database.service.gov.uk/) (<https://www.product-safety-database.service.gov.uk/>)

---

[Register of immigration advisers](https://portal.immigrationadviceauthority.gov.uk/s/)  
(<https://portal.immigrationadviceauthority.gov.uk/s/>)

---

[Request a basic DBS check](https://www.gov.uk/request-copy-criminal-record) (<https://www.gov.uk/request-copy-criminal-record>)

---

[Rural Payments Wales \(RPW\)](https://rpwonline.gov.wales/) (<https://rpwonline.gov.wales/>)

---

[Sign your mortgage deed](https://sign-your-mortgage-deed.landregistry.gov.uk/) (<https://sign-your-mortgage-deed.landregistry.gov.uk/>)

---

[Submit a barring referral \(<https://www.submit-a-barring-referral.service.gov.uk/start>\)](https://www.submit-a-barring-referral.service.gov.uk/start)

---

[Submit a General Aviation Report \(GAR\)  
\(<https://www.gov.uk/government/publications/general-aviation-operators-and-pilots-notification-of-flights>\)](https://www.gov.uk/government/publications/general-aviation-operators-and-pilots-notification-of-flights)

---

[Submit a National Security and Investment notification  
\(<https://www.gov.uk/government/collections/national-security-and-investment-act>\)](https://www.gov.uk/government/collections/national-security-and-investment-act)

---

[Submit a Pleasure Craft Report  
\(<https://www.gov.uk/guidance/submit-a-pleasure-craft-report>\)](https://www.gov.uk/guidance/submit-a-pleasure-craft-report)

---

[Submit an online APIS \(advance passenger information submission\) report  
\(<https://www.gov.uk/government/publications/providing-information-about-scheduled-aviation-flights>\)](https://www.gov.uk/government/publications/providing-information-about-scheduled-aviation-flights)

---

[Submit cosmetic product notifications \(<http://www.cosmetic-product-notifications.service.gov.uk>\)](http://www.cosmetic-product-notifications.service.gov.uk)

---

[Targeted retention incentives for further education teachers  
\(<https://www.gov.uk/government/collections/additional-payments-for-teaching-eligibility-and-payment-details>\)](https://www.gov.uk/government/collections/additional-payments-for-teaching-eligibility-and-payment-details)

---

[Use a lasting power of attorney \(<https://www.gov.uk/use-lasting-power-of-attorney>\)](https://www.gov.uk/use-lasting-power-of-attorney)

---



[Accessibility statement](#) [Cookies](#) [Terms and conditions](#)  
[Privacy notice](#) [Support \(opens in new tab\)](#)

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

# Using your GOV.UK One Login

## Contents

- Sign in to your GOV.UK One Login
- Proving your identity with GOV.UK One Login  
(/using-your-gov-uk-one-login/proving-your-identity)

# Sign in to your GOV.UK One Login

At the moment, you can only use GOV.UK One Login to access some government services. See the services you can use with GOV.UK One Login (<https://home.account.gov.uk/services-using-one-login>).

It does not work with all government accounts and services (/sign-in) yet (for example Universal Credit).

Over time, GOV.UK One Login will replace all other ways to sign in to services on GOV.UK, including Government Gateway.

This service is also available in Welsh (Cymraeg) (/defnyddio-eich-gov-uk-one-login).

If you need a GOV.UK One Login to use a service, and you do not already have one, you'll be able to create one when you first use that service.

## Sign in to:

- change your sign in details (email address, password or how you get security codes)
- see and access the services you've used with your GOV.UK One Login
- delete your GOV.UK One Login

**Sign in**

**Get help using GOV.UK One Login**

You can [contact the GOV.UK One Login team](https://home.account.gov.uk/contact-gov-uk-one-login) (<https://home.account.gov.uk/contact-gov-uk-one-login>) to get help, report a problem or give feedback.

You can also ask someone you know and trust to help you if you do not feel comfortable using GOV.UK One Login by yourself. Find out [what the person you ask can and cannot help you with](/guidance/help-someone-use-govuk-one-login) (</guidance/help-someone-use-govuk-one-login>).

## How GOV.UK One Login uses your information

To find out how your information is stored and used when you use GOV.UK One Login, see the [GOV.UK One Login privacy notice](/government/publications/govuk-one-login-privacy-notice) (</government/publications/govuk-one-login-privacy-notice>).

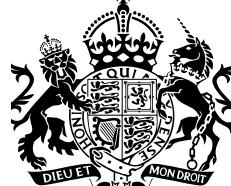
→ **Next**

[Proving your identity with GOV.UK One Login](#) ([/using-your-gov-uk-one-login/proving-your-identity](#))

[View a printable version of the whole guide](#) ([/using-your-gov-uk-one-login/print](#))



All content is available under the [Open Government Licence v3.0](#), except where otherwise stated



© Crown copyright

# Cookies on GOV.UK One Login

We'd like to use analytics cookies so we can understand how you use this website and make improvements.

[Accept analytics cookies](#)

[Reject analytics cookies](#)

[View cookies \(/cookies\)](#)



[About](#) [Documentation](#) [Get started](#) [Support](#) [Sign in](#)

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

[GOV.UK Services](#) > [GOV.UK One Login](#) > [About GOV.UK One Login](#)

[About GOV.UK One Login \(/about\)](#)

[Signing users in \(/about/signing-users-in\)](#)

[Checking users' identities \(/about/checking-users-identities\)](#)

**How users can prove their identity** [\(/about/checking-users-identities/evidence-types\)](#)

[The signed in experience \(/about/signed-in-experience\)](#)

[Roadmap](#) [\(/about/roadmap\)](#)

## How users can prove their identity

There are multiple ways we can check a user's identity. Users need to both:

- say who they are, for example by telling us their name, date of birth and recent address history
- prove that they are that person, for example making sure they look like their picture on their photo ID, or answering some security questions

## Evidence users can provide

The evidence a user can provide determines the route the user takes through GOV.UK One Login and how they prove their identity.

Users can currently prove their identity with GOV.UK One Login using their:

- web browser and the GOV.UK ID Check app
- web browser and at the Post Office
- web browser to answer security questions

# Using the GOV.UK ID Check app

Users can prove their identity using the GOV.UK ID Check app.

We'll check that:

- their ID documents are real
- they're a real person (also known as a 'liveness' check)
- they're the same person as in the document photos (also known as a 'likeness' check)

Users will need one of the following types of photo ID:

- UK or Northern Ireland photocard driving licence
- any passport with a biometric chip
- UK biometric residence permit (BRP)
- UK biometric residence card (BRC)
- UK Frontier Worker permit (FWP)

Users can use an expired BRP, BRC or FWP up to 18 months after its expiry date.

## Online and at a Post Office

Users will be asked to:

- enter details from their photo ID on GOV.UK
- go to a Post Office to have their photo ID scanned

Users will need one of the following types of photo ID:

- UK passport
- non-UK passport
- UK or Northern Ireland photocard driving licence
- European Union (EU) photocard driving licence
- national identity photocard from an EU country, Norway, Iceland or Liechtenstein

## Answering security questions online

Users answer security questions online (also known as knowledge-based verification questions) about things like their mobile phone contract and bank account.

The questions asked are based on their credit record and are answers only they should know.

Users can use one of the following types of photo ID:

- UK passport
- UK or Northern Ireland photocard driving licence

Users without photo ID can instead use their UK bank account details and HMRC tax record to prove their identity.

# Cookies on GOV.UK One Login

We'd like to use analytics cookies so we can understand how you use this website and make improvements.

[Accept analytics cookies](#)

[Reject analytics cookies](#)

[View cookies \(/cookies\)](#)



[About](#) [Documentation](#) [Get started](#) [Support](#) [Sign in](#)

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

[GOV.UK Services](#) > [GOV.UK One Login](#) > [About GOV.UK One Login](#)

[About GOV.UK One Login \(/about\)](#)

[Signing users in \(/about/signing-users-in\)](#)

[Checking users' identities \(/about/checking-users-identities\)](#)

[How users can prove their identity \(/about/checking-users-identities/evidence-types\)](#)

[The signed in experience \(/about/signed-in-experience\)](#)

[Roadmap \(/about/roadmap\)](#)

## Checking users' identities

You can use GOV.UK One Login to let users prove their identity, as well as [signing users in \(/about/signing-users-in\)](#).

Users will prove their identity once. They can reuse this proof to access other services that use GOV.UK One Login. This saves them time and effort.

### We currently provide a medium level of confidence

Our current journeys will give you a medium level of confidence in the user's identity, as defined by the government guidance on [how to prove and verify someone's identity \('GPG 45'\)](#).

(<https://www.gov.uk/government/publications/identity-proofing-and-verification-of-an-individual/how-to-prove-and-verify-someones-identity>)

It means using our identity checks will lower the risk of you accepting:

- completely made-up or 'synthetic' identities
- imposters who do not have a relationship with the claimed identity - for example, someone who has found

the claimed identity's information on social media

- imposters who have information about the claimed identity that's not in the public domain, for example, someone who works for the claimed identity's employer's HR department using information they have got to impersonate the claimed identity

We plan to offer low and high levels of confidence in the future.

## Deciding if this level is right for your service

To decide if this level of confidence is right for your service, you need to:

- identify the risks to your service
- check if having medium level of confidence in your users' identities will mitigate them
- consider how this will work with your users' needs

We know this is a complex area. We expect that most services will need to speak to us to confirm whether 'medium' is the right choice.

Our onboarding team is ready to answer your questions and work it out with you. You can get in touch using our [support form \(/contact-us\)](#) or [Slack channel](#) (<https://ukgovernmentdigital.slack.com/archives/C02AQUJ6WTC>).

## Identity data your service can get about your users

### User attributes

If a user successfully proves their identity, you'll always receive the following core identity information about them:

- their name
- date of birth
- the level of identity confidence

You can also ask for:

- postal addresses for the last 3 years (we check that addresses are genuine and if they're associated with

fraudulent activity)

- passport details, if the user proved their identity using their passport
- driving licence details, if the user proved their identity using their driving licence

We may ask you to provide a reason for requesting these pieces of information, so we can avoid unnecessary data sharing.

This is in addition to the data you'll get from the 'sign in' part of GOV.UK One Login, which is:

- a unique identifier
- an email address
- a mobile phone number, if the user set up two-factor authentication using their mobile phone number

If there's any other data your service needs - [get in touch \(/support\)](#) to talk to us about it.

## Audit and fraud data

If an identity has shown evidence of being fraudulent, we can provide more information about that user to you.

## Find out more

Read the [technical documentation \(<https://docs.sign-in.service.gov.uk/>\)](#) to see how to integrate with the identity checking part of GOV.UK One Login.

# Cookies on GOV.UK One Login

We'd like to use analytics cookies so we can understand how you use this website and make improvements.

[Accept analytics cookies](#)

[Reject analytics cookies](#)

[View cookies \(/cookies\)](#)



[About](#) [Documentation](#) [Get started](#) [Support](#) [Sign in](#)

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

[GOV.UK Services](#) > [GOV.UK One Login](#) > [About GOV.UK One Login](#)

[About GOV.UK One Login \(/about\)](#)

[Signing users in \(/about/signing-users-in\)](#)

[Checking users' identities \(/about/checking-users-identities\)](#)

[How users can prove their identity \(/about/checking-users-identities/evidence-types\)](#)

[The signed in experience \(/about/signed-in-experience\)](#)

[Roadmap \(/about/roadmap\)](#)

## Roadmap

Last updated: June 2025

### What we're working on now

**Users can add and remove two-factor authentication**  
SIGNED IN  
**methods** EXPERIENCE

Users will be able to add a backup two-factor authentication method or remove one they no longer need.

**Users can prove their identity with open banking**  
IDENTITY

Users will be able to prove their identity by signing in to their online banking and sharing information from their account.

### Recently released

GPG45 Low confidence route IDENTITY

Users can prove their identity to a GPG45 low confidence level.

## **Users can receive their Post Office customer letter by post** **IDENTITY**

Users who want to prove their identity at a Post Office can have their Post Office letter posted to them.

## **Using international addresses** **IDENTITY**

A user who lives at a non-UK address can use their non-UK address in GOV.UK One Login when proving their identity.

## **Users can prove their identity without photo ID** **IDENTITY**

Users who do not have a photo ID document can use their UK bank account details and their HMRC tax record to prove their identity.

## **User session ended when browser is closed** **SIGNED IN** **EXPERIENCE**

A user's session is ended when they quit their browser, lowering the risk of somebody else accessing their GOV.UK One Login on a shared device.

## **Email and phone number reputation checks** **SIGN IN**

Email and phone numbers are checked when a GOV.UK One Login is created, or email/phone numbers changed, to reduce the risk of fake GOV.UK One Logins being created.

## **Users can keep their details up to date after proving their identity** **SIGN IN**

Users can make updates to their identity details (for example, their name) to keep their GOV.UK One Login up to date.

## **Updates**

As we're still at an early stage, our plans may shift. We'll update this page when this happens and add more detail

when we can.

We'll keep sharing in [blog posts](#) (<https://gds.blog.gov.uk/category/govuk-onelogin/>) and at our regular cross-government show and tells too.

[Join our mailing list \(/mailing-list\)](#) to stay up to date.

# Cookies on GOV.UK One Login

We'd like to use analytics cookies so we can understand how you use this website and make improvements.

[Accept analytics cookies](#)

[Reject analytics cookies](#)

[View cookies \(/cookies\)](#)



[About](#) [Documentation](#) [Get started](#) [Support](#) [Sign in](#)

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

[GOV.UK Services](#) > [GOV.UK One Login](#) > [About GOV.UK One Login](#)

[About GOV.UK One Login \(/about\)](#)

[Signing users in \(/about/signing-users-in\)](#)

[Checking users' identities \(/about/checking-users-identities\)](#)

[How users can prove their identity \(/about/checking-users-identities/evidence-types\)](#)

[The signed in experience \(/about/signed-in-experience\)](#)

[Roadmap \(/about/roadmap\)](#)

## The signed in experience

When a user creates a GOV.UK One Login to access a service, they also get a space where they can manage their details and see other services they've used.

Users can be signed in to this space and access it whether they're signed in to your service or not.

## What users can do in their GOV.UK One Login

### Update credentials

Users can change their:

- email address
- password
- phone number

We've got some recommended content to help you [show users where to change their GOV.UK One Login credentials \(/documentation/design-recommendations/change-credentials\)](#).

[Delete their GOV.UK One Login](#)

Users can delete their GOV.UK One Login.

If they do this, we'll delete their records from all our live service databases within GOV.UK One Login, including their:

- sign in details
- saved identity information

They'll no longer be able to use their GOV.UK One Login to access your service.

We'll keep some data about what the user has done with their GOV.UK One Login for auditing and monitoring reasons.

## **See and access their services in one place**

When a user signs in to your service with their GOV.UK One Login, a link to your service or to your GOV.UK start page appears in the ‘Your services’ section.

This means they can easily access all the services they've used with their GOV.UK One Login from one place.

Users can continue to access your service through familiar routes, like landing on a GOV.UK start page.

## **See a user's GOV.UK One Login space**

Here are the two main pages that make up a user's GOV.UK One Login space. They can get to this space if you link to it from your service, following our [design recommendations for letting users change credentials](#) (</documentation/design-recommendations/change-credentials>).

[Your services](#) [Security](#) [Sign out](#)**BETA** This is a new service – your [feedback](#) will help us improve it.

## Your services

You're signed in as [name@email.com](#)

### Your accounts

#### GOV.UK email subscriptions

See and manage the updates you get about GOV.UK pages you're interested in.

[Go to your GOV.UK email subscriptions](#)

Last used: 10 October 2022

### Other services you've used

[Sign your mortgage deed](#)

Last used: 14 May 2018

[Request a basic DBS check](#)

Last used: 14 May 2018

#### Services you can use with GOV.UK One Login

GOV.UK One Login is new. At the moment you can only use it to access some government services.

► [Services you can use with GOV.UK One Login](#)

GOV.UK One Login does not work with all government accounts and services yet (for example Government Gateway or Universal Credit).

In the future, you'll be able to use GOV.UK One Login to access all services on GOV.UK.

[Your services](#) [Security](#) [Sign out](#)**BETA** This is a new service – your [feedback](#) will help us improve it.

## Security

#### Your sign in details

##### Email address

name@email.com

[Change](#)

##### Password

.....

[Change](#)

#### How you get security codes

We use security codes to make sure it's you when you sign in.

##### Test message

Phone number ending with 0406

[Change](#)

#### Delete your GOV.UK One Login

This will permanently delete your GOV.UK One Login. You'll no longer be able to access the services you've used with it.

[Delete your GOV.UK One Login](#)

## Security page example

## Your services page example

# Cookies on GOV.UK One Login

We'd like to use analytics cookies so we can understand how you use this website and make improvements.

[Accept analytics cookies](#)

[Reject analytics cookies](#)

[View cookies \(/cookies\)](#)



[About](#) [Documentation](#) [Get started](#) [Support](#) [Sign in](#)

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

[GOV.UK Services](#) > [GOV.UK One Login](#) > [About GOV.UK One Login](#)

[About GOV.UK One Login \(/about\)](#)

[Signing users in \(/about/signing-users-in\)](#)

[Checking users' identities \(/about/checking-users-identities\)](#)

[How users can prove their identity \(/about/checking-users-identities/evidence-types\)](#)

[The signed in experience \(/about/signed-in-experience\)](#)

[Roadmap \(/about/roadmap\)](#)

## Signing users in

You can use GOV.UK One Login to help your users sign into your service quickly and easily. This is also called ‘authentication’.

This means you know it’s always the same person accessing the service.

## The user experience

You can let users create their GOV.UK One Login or sign in at:

- the start of your service, if you need them all to have accounts
- the point where they want to save their progress and come back later, if you want to add this option to a complex journey

Users can create a GOV.UK One Login with their email address and password.

You can request that your users also use two-factor authentication. They have two options:

- getting text messages containing security codes sent to their UK or international mobile phone number
- using security codes generated by an authenticator app

They can use these details whenever they need to sign in to your service.

If they forget their details, they can recover them.

When a user creates a GOV.UK One Login, they'll also get access to a space where they can manage their details and see the services they've used. [Find out more about what's available in a user's GOV.UK One Login \(/about/signed-in-experience\)](#) when they're signed in.

## See the sign in user journeys

[View journey maps of the sign in journey. \(/documentation/user-journeys\)](#)

# Cookies on GOV.UK One Login

We'd like to use analytics cookies so we can understand how you use this website and make improvements.

[Accept analytics cookies](#)

[Reject analytics cookies](#)

[View cookies \(/cookies\)](#)



[About](#) [Documentation](#) [Get started](#) [Support](#) [Sign in](#)

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

[GOV.UK Services](#) > [GOV.UK One Login](#)

[About GOV.UK One Login  
\(/about\)](#)

[Signing users in  
\(/about/signing-users-in\)](#)

[Checking users' identities  
\(/about/checking-users-identities\)](#)

[How users can prove their identity  
\(/about/checking-users-identities/evidence-types\)](#)

[The signed in experience  
\(/about/signed-in-experience\)](#)

[Roadmap \(/about/roadmap\)](#)

## About GOV.UK One Login

GOV.UK One Login lets your users sign in and prove their identity so they can access your service quickly and easily.

It will ultimately allow users to access any government service using the same email address and password, and also only need to prove their identity once.

## Why use GOV.UK One Login

Using GOV.UK One Login means:

- you will not need to build and maintain your own sign in and identity checking systems
- you save money as GOV.UK One Login is centrally funded and free for government services to use
- a reduction in fraud thanks to our dedicated fraud prevention team and the built in protection provided by our [GPG45](#) (<https://www.gov.uk/government/publications/identity-proofing-and-verification-of-an-individual>) compliant systems
- improved accessibility, as GOV.UK One Login is based on common, accessible components from the GOV.UK Design System

# Who can use GOV.UK One Login

GOV.UK One Login is currently available to all central government departments and agencies.

Other organisations may also be able to use GOV.UK One Login. [Get in touch \(<https://www.sign-in.service.gov.uk/support>\)](https://www.sign-in.service.gov.uk/support) and we'll have a chat to find out more about your service.

## What GOV.UK One Login offers

### Users can sign in

Users can sign in to your service using GOV.UK One Login, with their username, password and two-factor authentication.

You can add this sign in functionality into your service on its own. You do not have to use the identity checks functionality if you do not need it.

[Find out more about signing users in \(/about/signing-users-in\).](#)

### Users can prove their identity

Users can prove their identity so you know they are who they say they are.

If you want to add identity checks, you must also use GOV.UK One Login to sign in your users.

[Find out more about checking users' identities \(/about/checking-users-identities\)](#)

## Try GOV.UK One Login

You can:

- [set up our integration environment \(<https://admin.sign-in.service.gov.uk/sign-in/enter-email-address>\)](#) to see how GOV.UK One Login works in your service
- [request access to our HTML prototype \(<https://www.sign-in.service.gov.uk/documentation/end-to-end-prototype/identity-journeys>\)](#) to explore journeys within GOV.UK One Login

Also, read our [technical documentation \(<https://docs.sign-in.service.gov.uk/how-gov-uk-one-login-works/#how-gov-uk-one->](#)

[login-works](#)) and [design recommendations](#) (<https://www.sign-in.service.gov.uk/documentation/design-recommendations>) to find out more about how GOV.UK One Login works.

## GOV.UK Wallet

GOV.UK Wallet will allow users to store government-issued documents on their phones.

You will be able to use GOV.UK Wallet to:

- allow a user to store a digital version of a document you produce, for example a driving licence
- request information you need to know about a user as part of your service

Find out more about [GOV.UK Wallet](#) (<https://www.gov.uk/guidance/using-govuk-wallet-in-government>).

## Support

Support for your service is available during office hours using our [support form](#) (<https://www.sign-in.service.gov.uk/support>) or [Slack channel](#) (<https://ukgovernmentdigital.slack.com/archives/C02AQUJ6WTC>).

Our technical team is also available 24/7 for urgent incidents.

## How to start using GOV.UK One Login

If you think GOV.UK One Login might be right for your service, [register your interest](#) (<https://www.sign-in.service.gov.uk/register>).

We'll contact you within 5 days to find out more about what your service needs.

# Cookies on GOV.UK One Login

We'd like to use analytics cookies so we can understand how you use this website and make improvements.

[Accept analytics cookies](#)

[Reject analytics cookies](#)

[View cookies \(/cookies\)](#)



One Login Beta

About Documentation Get started Support Sign in

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

[GOV.UK Services](#) > [GOV.UK One Login](#)

> [Accessibility statement for the GOV.UK One Login website](#)

## Accessibility statement for the GOV.UK One Login website

GOV.UK One Login and the GOV.UK One Login admin tool are part of the wider GOV.UK website. There's a separate accessibility statement for the main GOV.UK website.

This accessibility statement only contains information about the [GOV.UK One Login website \(/\)](#) and the [GOV.UK One Login admin tool](#).

This accessibility statement explains:

- how accessible the website is
- what work we're planning to do
- what to do if you have difficulty using it
- how to report any accessibility problems

## What this website should do

This website was created by the Government Digital Service. It is designed to be used by as many people as possible.

The text should be clear and simple to understand.

You should be able to:

- zoom in up to 300% without problems
- navigate most of the website using just a keyboard
- navigate most of the website using speech recognition software
- use most of the website using a screen reader (including the most recent versions of JAWS, NVDA and VoiceOver)

## How we tested this website

The GOV.UK One Login website was last tested in July 2022. The test was carried out by the Digital Accessibility Centre (DAC), who produced an accessibility audit report on 13 October 2021.

DAC assessed the GOV.UK One Login website against the Web Content Accessibility Guidelines (WCAG) 2.1.

The GOV.UK One Login admin tool was last tested in October 2022 by the GOV.UK One Login programme. We carried out automated and manual testing against the Web Content Accessibility Guidelines WCAG 2.1.

## What we're doing to improve accessibility

Following recommendations from the testing carried out by the DAC in October 2021, we fixed outstanding A and AA issues on the GOV.UK One Login site.

This included:

- making link text more descriptive
- labelling form fields correctly
- fixing some ‘skip’ links that did not take users to the correct destination

On the GOV.UK One Login admin tool, following our own internal testing, we have fixed outstanding A and AA issues.

This included:

- making it clear when a user is activating, hovering or focusing over the ‘Show password’ button

- adding a linked error message and error summary to the Redirect URI field on the ‘Change your redirect URIs’ page

We are continuing to look at fixing the accessibility issues on the [GOV.UK One Login's status page](https://status.account.gov.uk/) (<https://status.account.gov.uk/>) which was added in May 2023. You can [find out more about the non-compliances on the GOV.UK One Login status page](#).

## What to do if you have difficulty using this website

If you have difficulty using this website, contact us by sending an email to [govuk-sign-in@digital.cabinet-office.gov.uk](mailto:govuk-sign-in@digital.cabinet-office.gov.uk)

## Reporting accessibility problems

We’re always looking to improve the accessibility of this website.

If you find any problems, or think we’re not meeting accessibility requirements, email [govuk-sign-in@digital.cabinet-office.gov.uk](mailto:govuk-sign-in@digital.cabinet-office.gov.uk)

## If you’re not happy with our response

The Equality and Human Rights Commission (EHRC) is responsible for enforcing the Public Sector Bodies (Websites and Mobile Applications) (No. 2) Accessibility Regulations 2018 (the ‘accessibility regulations’).

If you are not happy with how we respond to your complaint, contact the [Equality Advisory and Support Service](https://www.equalityadvisoryservice.com/) (<https://www.equalityadvisoryservice.com/>), which is run on behalf of EHRC.

## Technical information about this website’s accessibility

The Government Digital Service is committed to making its websites accessible, in accordance with the Public Sector Bodies (Websites and Mobile Applications) (No. 2) Accessibility Regulations 2018.

This website is partially compliant with the WCAG version 2.1 AA standard, due to the non-compliances listed below.

## Non-compliance with the accessibility regulations

The following content on [GOV.UK One Login's status page](https://status.account.gov.uk/) (<https://status.account.gov.uk/>) is not compliant with the WCAG version 2.1 AA standard.

### Level A failures

- Text presentation uses incorrect semantics - for example headers and other elements of the visual structure are not accessible to people who cannot see this visual presentation (WCAG success criterion 1.3.1)
- Some links are implemented in a way that is not consistent with HTML nesting rules - links may not work or may behave in unexpected ways across different assistive technologies (WCAG success criterion 4.1.1)

### Level AA failures

- Multiple text elements use poor contrast with their backgrounds making some text elements not perceivable to people with moderately low vision (WCAG success criterion 1.4.3)
- Text size cannot be increased to 200% without negative effects such as text overlapping or text being cut off (WCAG success criterion 1.4.4)
- Content cannot be presented without loss of information when using browser zoom at 400% (WCAG success criterion 1.4.10)
- ‘Subscribe to updates’ forms, when submitted with incorrect data, do not inform screen reader users of the error message when it becomes available (WCAG success criterion 4.1.3)

Together with the supplier of the GOV.UK One Login status page, we intend to make this service fully accessible. We are continuously improving and reviewing the service.

## Preparation of this accessibility statement

This statement was prepared on 16 May 2023. It was last reviewed on 18 October 2023.



# Cookies on GOV.UK One Login

We'd like to use analytics cookies so we can understand how you use this website and make improvements.

[Accept analytics cookies](#)

[Reject analytics cookies](#)

[View cookies \(/cookies\)](#)



About Documentation Get started Support Sign in

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

< [Back](#)

## Contact us

You can find out how to:

- [get started \(/getting-started\)](#) with GOV.UK One Login
- [register your interest \(/register\)](#) in using GOV.UK One Login

Or if you still have questions, you can contact us using our [support form \(/contact-us\)](#) or via our [Slack channel](#) (<https://ukgovernmentdigital.slack.com/archives/C02AQUJ6WTC>).

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

# Cookies notice for the GOV.UK One Login website

GOV.UK One Login puts small files (known as ‘cookies’) on to your computer.

Cookies are used to remember the notifications you’ve seen so we don’t show them again.

You’ll normally see a message on the site before we store a cookie on your computer.

Find out [how to manage cookies \(https://ico.org.uk/for-the-public/online/cookies/\)](https://ico.org.uk/for-the-public/online/cookies/).

## Our introductory message

You may see a pop-up welcome message when you first visit the GOV.UK One Login website. This message will ask whether you want to accept or reject analytics cookies. Once you’ve saved your preferences, we’ll store a cookie on your computer to remember them.

You can change your cookie settings at any time from this page.

Name	Purpose	Expires
cookies_preferences_set	Saves your cookie consent settings	1 year

## Analytics cookies (optional)

With your permission, we use Google Analytics to collect data about how you use the GOV.UK One Login website. This information helps us to improve the website.

Google is not allowed to use or share our analytics data with anyone.

Google Analytics stores anonymised information about:

- how you got to the GOV.UK One Login website
- the pages you visit
- how long you spend on each page
- what you click on while you're visiting the site
- the device and browser you're using

Name	Purpose	Expires
_ga	Checks if you've visited the GOV.UK One Login website before. This helps us count how many people visit our site.	2 years
_gid	Checks if you've visited the GOV.UK One Login website before. This helps us count how many people visit our site.	24 hours
_gat_gtag_[property_id]	This helps us track how you use our site, for example how long you spend on a page.	1 minute

## Do you want to accept analytics cookies?



Yes



No

[Save cookie settings](#)

---

# Cookies on GOV.UK One Login

We'd like to use analytics cookies so we can understand how you use this website and make improvements.

[Accept analytics cookies](#)

[Reject analytics cookies](#)

[View cookies \(/cookies\)](#)



[About](#) [Documentation](#) [Get started](#) [Support](#)  
[Sign in](#)

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

[GOV.UK Services](#) > [GOV.UK One Login](#) > [Documentation](#)

[Technical documentation \(/documentation\)](#)

[Sign in user journey maps \(/documentation/user-journeys\)](#)

[Proving identity journey maps \(/documentation/identity-journeys\)](#)

[GOV.UK One Login prototype \(/documentation/end-to-end-prototype/identity-journeys\)](#)

[Design recommendations \(/documentation/design-recommendations\)](#)

[Let users create a GOV.UK One Login to save progress \(/documentation/design-recommendations/save-progress\)](#)

[Let users navigate to their GOV.UK One Login and sign out easily \(/documentation/design-recommendations/let-users-navigate-sign-out\)](#)

## User groups who may find it harder to prove their identity

GOV.UK One Login lets your users sign in and prove their identity so they can access your service easily.

### Sign in

To allow your users to sign in with GOV.UK One Login, they'll need:

- an email address
- access to a mobile phone or an authenticator app (mobile or browser based)

### Identity proving

There are 3 routes for users to prove their identity using GOV.UK One Login:

- with the GOV.UK ID Check app
- by answering security questions in the browser
- a combination of online and at a Post Office

Show users where to change their GOV.UK One Login credentials  
(/documentation/design-recommendations/change-credentials)

---

Moving to GOV.UK One Login if your service already proves users identities  
(/documentation/design-recommendations/migrating-users)

---

Moving to GOV.UK One Login if you need to sign in your users  
(/documentation/design-recommendations/migrating-sign-in-pattern)

---

What to say about GOV.UK One Login on start pages  
(/documentation/design-recommendations/start-page)

---

How to talk about GOV.UK One Login  
(/documentation/design-recommendations/how-to-talk-about)

---

Prepare your users to move to GOV.UK One Login  
(/documentation/design-recommendations/prepare-to-move)

---

Business users and GOV.UK One Login  
(/documentation/design-recommendations/business-users)

---

**User groups who may find it harder to prove their identity**  
(/documentation/design-recommendations/barriers-to-proving-identity)

---

Users will be guided to the best route for them to use to prove their identity.

## Users who may struggle to prove their identity

Currently, GOV.UK One Login only lets users prove their identity using certain documents or information. There are also certain sections of the user journey that can only be completed online. This means it does not work for everyone yet. There are certain user groups that this will impact more heavily.

You can look at the results of the [user segmentation survey](https://www.gov.uk/government/publications/govuk-one-login-user-segmentation-survey-summary) (<https://www.gov.uk/government/publications/govuk-one-login-user-segmentation-survey-summary>) we ran to understand more about what barriers could affect people when trying to prove their identity with GOV.UK One Login.

### Children under the age of 17

Children under the age of 17 are likely to struggle as they will not have a credit history or digital footprint with government, which means they will not be able to answer any security questions. This will make it harder for them to prove their identity.

In future, we plan to introduce new product features to make it easier for them to prove their identity.

### UK citizens living abroad

This group can prove their identity using the GOV.UK ID Check app and a chipped passport.

They will not be able to prove their identity using other routes, unless they can visit a UK Post Office.

### Non-UK citizens living in the UK

This group's likelihood of successfully proving their identity depends on:

- how long they've been in the UK
- what identity documents they have available to them
- if they have a permanent UK address

## Non-UK citizens living overseas

This group can only use the GOV.UK ID Check app to prove their identity. They'll also need a chipped passport.

## Users with a small digital or financial footprint

Users with a limited credit history or digital footprint with government could struggle to prove their identity, especially if trying to answer security questions.

If your service has a large number of users in any of these groups, [get in touch \(<https://www.sign-in.service.gov.uk/support>\)](https://www.sign-in.service.gov.uk/support).

We can advise:

- about other ways you can help those users prove their identity, for example by you providing a paper-based route or offering more contact centre support
- if now is the right time for you to start using GOV.UK One Login

If a large number of your users are likely to struggle to prove their identity, you'll need to provide another way for those users to access your service. If you do not have this, [get in touch \(<https://www.sign-in.service.gov.uk/support>\)](https://www.sign-in.service.gov.uk/support) so we can discuss if GOV.UK One Login is right for your service.

# Cookies on GOV.UK One Login

We'd like to use analytics cookies so we can understand how you use this website and make improvements.

[Accept analytics cookies](#)

[Reject analytics cookies](#)

[View cookies \(/cookies\)](#)



[About](#) [Documentation](#) [Get started](#) [Support](#)  
[Sign in](#)

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

[GOV.UK Services](#) > [GOV.UK One Login](#) > [Documentation](#)

[Technical documentation \(/documentation\)](#)

[Sign in user journey maps \(/documentation/user-journeys\)](#)

[Proving identity journey maps \(/documentation/identity-journeys\)](#)

[GOV.UK One Login prototype \(/documentation/end-to-end-prototype/identity-journeys\)](#)

[Design recommendations \(/documentation/design-recommendations\)](#)

[Let users create a GOV.UK One Login to save progress \(/documentation/design-recommendations/save-progress\)](#)

[Let users navigate to their GOV.UK One Login and sign out easily \(/documentation/design-recommendations/let-users-navigate-sign-out\)](#)

## Business users and GOV.UK One Login

GOV.UK One Login lets your users sign in and prove their identity so they can access your service easily. It was designed for citizens, but can also work for business users.

### What we mean by business users

When we talk about business users, we mean services that have:

- users who need GOV.UK One Login to carry out tasks in their work life, for example Companies House users
- internal users, for example civil servants

### What GOV.UK One Login offers

GOV.UK One Login will help you sign in your users, and prove their identity so you know they are who they claim to be.

GOV.UK One Login does not tie that user to a particular role or organisation.

Show users where to change their GOV.UK One Login credentials  
(/documentation/design-recommendations/change-credentials)

---

Moving to GOV.UK One Login if your service already proves users identities  
(/documentation/design-recommendations/migrating-users)

---

Moving to GOV.UK One Login if you need to sign in your users  
(/documentation/design-recommendations/migrating-sign-in-pattern)

---

What to say about GOV.UK One Login on start pages  
(/documentation/design-recommendations/start-page)

---

How to talk about GOV.UK One Login  
(/documentation/design-recommendations/how-to-talk-about)

---

Prepare your users to move to GOV.UK One Login  
(/documentation/design-recommendations/prepare-to-move)

---

Business users and GOV.UK One Login  
(/documentation/design-recommendations/business-users)

---

User groups who may find it harder to prove their identity  
(/documentation/design-recommendations/barriers-to-proving-identity)

---

## What GOV.UK One Login does not offer for business users

GOV.UK One Login will not:

- prove whether a user belongs to a particular organisation - or perform a certain role - that would still need to be done by the service (this is sometimes called 'eligibility')
- separate a user's business uses for GOV.UK One Login from their personal ones unless the user chooses to create a separate GOV.UK One Login for work purposes, using a different email address

## Email address as username

A user's GOV.UK One Login username is always an email address. The only time we use this email address to communicate with the user is if they want to make changes to their GOV.UK One Login. For example, if a user wants to change the phone number they use for 2-factor authentication.

When choosing an email address for GOV.UK One Login it's important that it's an email the user is likely to have long term access to. Or, at least, for as long as the user will need to access the service associated with that GOV.UK One Login.

The same applies for choosing a mobile phone number to use with GOV.UK One Login. It can be a personal phone number or a work one, but the user will need long-term access to it. If a user leaves an organisation and loses access to their work phone, they will not be able to receive security codes to sign in to their GOV.UK One Login.

## What you need to consider

If users do use a personal email address for your service, you need to consider:

- your service's preference may be for users to access your service using a work or organisation email address
- whether your service is dependant on users coming from a professional domain, for example to help establish eligibility to use your service

- users may not be comfortable using a personal email address for work purposes

Also, users may not want to use their personal mobile phone for 2-factor authentication. This will be needed when a user sets up their GOV.UK One Login, and depending on how you choose to implement GOV.UK One Login, potentially every time they sign in.

## Contacting your users

As email address is essentially just a username for GOV.UK One Login, we will not know if the user still has access to that address unless they tell us.

If you want to use that email address for correspondence, you may want to confirm with the users that's the best way to contact them.

## Find out more

If you're interested in knowing more about GOV.UK One Login and business users, [get in touch. \(\)](https://www.sign-in.service.gov.uk/support)

# Cookies on GOV.UK One Login

We'd like to use analytics cookies so we can understand how you use this website and make improvements.

[Accept analytics cookies](#)

[Reject analytics cookies](#)

[View cookies \(/cookies\)](#)



[About](#) [Documentation](#) [Get started](#) [Support](#)  
[Sign in](#)

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

[GOV.UK Services](#) > [GOV.UK One Login](#) > [Documentation](#) > [Design recommendations](#)

[Technical documentation \(/documentation\)](#)

[Sign in user journey maps \(/documentation/user-journeys\)](#)

[Proving identity journey maps \(/documentation/identity-journeys\)](#)

[GOV.UK One Login prototype \(/documentation/end-to-end-prototype/identity-journeys\)](#)

[Design recommendations \(/documentation/design-recommendations\)](#)

[Let users create a GOV.UK One Login to save progress \(/documentation/design-recommendations/save-progress\)](#)

[Let users navigate to their GOV.UK One Login and sign out easily \(/documentation/design-recommendations/let-users-navigate-sign-out\)](#)

## Show users where to change their GOV.UK One Login credentials

Your users need to be able to navigate from your service to their GOV.UK One Login so they can check or update their sign in details.

[We recommend that you use the GOV.UK One Login service header for this \(/documentation/design-recommendations/let-users-navigate-sign-out\).](#)

There are 2 other patterns we've developed and tested which can be used if:

- you cannot use the GOV.UK One Login service header
- you need to give your users additional routes to change their sign in details

## Contents

- [Option 1: the GOV.UK One Login service header](#)
- [Option 2: a paragraph on a dedicated page](#)
- [Option 3: inset text on your service homepage](#)

Show users where to change their GOV.UK One Login credentials  
(/documentation/design-recommendations/change-credentials)

---

Moving to GOV.UK One Login if your service already proves users identities  
(/documentation/design-recommendations/migrating-users)

---

Moving to GOV.UK One Login if you need to sign in your users  
(/documentation/design-recommendations/migrating-sign-in-pattern)

---

What to say about GOV.UK One Login on start pages  
(/documentation/design-recommendations/start-page)

---

How to talk about GOV.UK One Login  
(/documentation/design-recommendations/how-to-talk-about)

---

Prepare your users to move to GOV.UK One Login  
(/documentation/design-recommendations/prepare-to-move)

---

Business users and GOV.UK One Login  
(/documentation/design-recommendations/business-users)

---

User groups who may find it harder to prove their identity  
(/documentation/design-recommendations/barriers-to-proving-identity)

---

## Option 1: the GOV.UK One Login service header

Use [the GOV.UK One Login service header](#)  
(/documentation/design-recommendations/let-users-navigate-sign-out) if your service is using GOV.UK One Login.

The header gives users an easy, consistent route from your service to their GOV.UK One Login and a way to sign out. Your users can also update their sign in details or change how they get security codes.

---

## Option 2: a paragraph on a dedicated page

You can use this paragraph (written here in [Markdown](#) (<https://www.markdownguide.org/basic-syntax/>)):

You use your GOV.UK One Login to sign in to {Service name}.

You can change these details in your GOV.UK One Login:

- email address
- password
- how you get security codes to sign in

[Change your sign in details in your GOV.UK One Login]  
(<https://home.account.gov.uk/settings>)

[Home](#) [Your profile](#) [Sign out](#)

## Your profile

### Service details

### Sign in details

You use your GOV.UK One Login to sign in to [service name].

You can change these details in your GOV.UK One Login:

- email address
- password
- how you get security codes to sign in

[Change your sign in details in your GOV.UK One Login](#)

Menu ▾

[Home](#)[Your profile](#)[Change your sign in details](#)[Your messages](#)[Sign out](#)

## Change your sign in details

You use your GOV.UK One Login to sign in to [service name].

You can change these details in your GOV.UK One Login:

- email address
- password
- how you get security codes to sign in

[Change your sign in details in your GOV.UK One Login](#)

## Example A

## Example B

### What is it?

A paragraph of text which explains that users use GOV.UK One Login to sign in to this service. It tells users to go to their GOV.UK One Login to update their sign in details.

On example A, this text is on a ‘Your profile’ page. It’s underneath a section for ‘Service details’, which could be other ‘profile’ information your service collects.

On example B, the text is the only information on a ‘Change your sign in details’ page.

### When to use it

Use this pattern if your service has a dedicated section where users can manage details and personal information.

Choose a heading from the examples above based on which option is grammatically consistent with the rest of the headings on the page.

For example, use ‘Change your sign in details’ if all your other headings are actions, like ‘View my profile’.

Use ‘Sign in details’ if they’re all nouns, like ‘Service details’.

## How it works

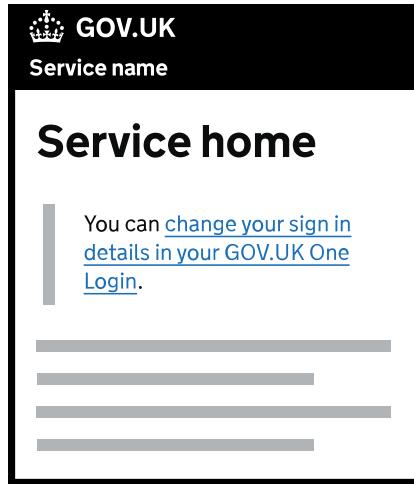
The ‘Change your sign in details in your GOV.UK One Login’ link will take users to the page in their GOV.UK One Login where they can change their details.

---

## Option 3: inset text on your service homepage

You can use this paragraph:

You can [change your sign in details in your GOV.UK One Login]  
(<https://home.account.gov.uk/settings>).



## What is it?

Content in an [inset text component](https://design-system.service.gov.uk/components/inset-text/) (<https://design-system.service.gov.uk/components/inset-text/>) tells users they can change their sign in details in their GOV.UK One Login.

## When to use it

Use this design if your service is simple and does not have a dedicated area for users to manage their details.

In the example, we've put it on the service homepage. You can choose wherever makes most sense for your service though.

## How it works

The link will take users to the page in their GOV.UK One Login where they can change their details.

# Cookies on GOV.UK One Login

We'd like to use analytics cookies so we can understand how you use this website and make improvements.

[Accept analytics cookies](#)

[Reject analytics cookies](#)

[View cookies \(/cookies\)](#)



[About](#) [Documentation](#) [Get started](#) [Support](#)  
[Sign in](#)

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

[GOV.UK Services](#) > [GOV.UK One Login](#) > [Documentation](#)

[Technical documentation \(/documentation\)](#)

[Sign in user journey maps \(/documentation/user-journeys\)](#)

[Proving identity journey maps \(/documentation/identity-journeys\)](#)

[GOV.UK One Login prototype \(/documentation/end-to-end-prototype/identity-journeys\)](#)

[Design recommendations \(/documentation/design-recommendations\)](#)

[Let users create a GOV.UK One Login to save progress \(/documentation/design-recommendations/save-progress\)](#)

[Let users navigate to their GOV.UK One Login and sign out easily \(/documentation/design-recommendations/let-users-navigate-sign-out\)](#)

## How to talk about GOV.UK One Login

### When talking to users

GOV.UK One Login allows you to sign in to some government services using the same email address and password.

In the future you'll be able to use your GOV.UK One Login to access all services on GOV.UK.

### Using the name in content

Always use the product's full name - 'GOV.UK One Login'.

Examples:

- GOV.UK One Login is new. At the moment you can only use a / your GOV.UK One Login with a few services
- Give feedback about GOV.UK One Login

Never use:

- GUOL

Show users where to change their GOV.UK One Login credentials  
(/documentation/design-recommendations/change-credentials)

---

Moving to GOV.UK One Login if your service already proves users identities  
(/documentation/design-recommendations/migrating-users)

---

Moving to GOV.UK One Login if you need to sign in your users  
(/documentation/design-recommendations/migrating-sign-in-pattern)

---

What to say about GOV.UK One Login on start pages  
(/documentation/design-recommendations/start-page)

---

**How to talk about GOV.UK One Login**  
(/documentation/design-recommendations/how-to-talk-about)

---

Prepare your users to move to GOV.UK One Login  
(/documentation/design-recommendations/prepare-to-move)

---

Business users and GOV.UK One Login  
(/documentation/design-recommendations/business-users)

---

User groups who may find it harder to prove their identity  
(/documentation/design-recommendations/barriers-to-proving-identity)

---

- GOV.UK OL
- One Login (without GOV.UK logo)
- GOV.UK One Sign In / Sign On
- GOV.UK One Log In / Log On
- One Login for government
- Single sign-on for government

## In Welsh

Do not translate the name GOV.UK One Login into Welsh.

## GOV.UK ID Check app

Use the full name of the app when referring to it: 'GOV.UK ID Check app'.

## In products

Use GOV.UK One Login as a noun.

This helps with account confusion because many services have accounts. For example, Childcare account, Personal tax account.

This allows us to categorise and label relevant services as accounts in a user's GOV.UK One Login home.

## When we know users have a GOV.UK One Login

Examples:

- Use / Use your GOV.UK One Login to...
- You can...with / with your GOV.UK One Login
- You already have a GOV.UK One Login
- You can currently only prove your identity with your GOV.UK One Login if you have a UK passport
- There are some government services that you cannot use with your GOV.UK One Login yet

## When we don't know if users have a GOV.UK One Login

Examples:

- Use / Use your GOV.UK One Login to...
- You can...with / with your GOV.UK One Login

- If you already have a GOV.UK One Login
- You can currently only prove your identity with / with a GOV.UK One Login if you have a UK passport or driving licence
- There are some government services that you cannot use with / with a GOV.UK One Login yet

## Using the word 'account'

Do not use the word ‘account’ to describe or refer to GOV.UK One Login. Instead say:

- Use / Use your GOV.UK One Login to...
- You already have a GOV.UK One Login

## In headers and banners

### Desktop

Always use GOV.UK One Login.

Never use:

- GOV.UK One Login account
- One Login

## Creating a GOV.UK One Login

For creating a GOV.UK One Login, use:

- Create a GOV.UK One Login to...
- You need a GOV.UK One Login to continue

## Signing in

For signing in to a service, use:

- Sign in with / with your / with a GOV.UK One Login
- Use GOV.UK One Login to sign in, if you have one

For signing in to GOV.UK One Login, use:

- Sign in to your GOV.UK One Login
- If you already use / have a GOV.UK One Login, you can sign in to manage your settings.

Never use:

- Login to GOV.UK One Login

## Getting help and support

Example:

If you need help, contact the GOV.UK One Login team.

## Identity

### Identity proving

Identity proving is the act of confirming that an identity:

- is a real person
- actually belongs to the person trying to prove their identity

A service needs to check a user's identity if, for example, it:

- shows a user personal information about themselves, such as their driving licence or passport details
- gives the user something valuable, such as money or benefits

Use 'Prove your identity' when it's someone proving their own identity.

Use 'proved' and not 'proven'. For example, 'You have already proved your identity'.

Never use:

- identity verification
- identity checking

### Prove your identity in person

This is a way for users to prove their identity in person, without using the online service.

Example:

You can prove your identity in person

Never use:

- in-person verification (IPV)

## Answering security questions

This is a way to prove someone is who they claim to be by asking them questions only they should know the answers to.

We use knowledge-based verification (KBV) questions as part of our identity journey.

Do not use this term when talking about GOV.UK One Login. Instead, talk about ‘answering security questions using information only you should know’.

Never use:

- knowledge-based verification (KBV)

## Further guidance about GOV.UK One Login

You could also look at the:

- GOV.UK One Login section in the [GOV.UK style guide](https://www.gov.uk/guidance/style-guide/a-to-z-of-gov-uk-style#govuk-one-login) (<https://www.gov.uk/guidance/style-guide/a-to-z-of-gov-uk-style#govuk-one-login>)
- [Using your GOV.UK One Login guide](https://www.gov.uk/using-your-gov-uk-one-login) (<https://www.gov.uk/using-your-gov-uk-one-login>)

# Cookies on GOV.UK One Login

We'd like to use analytics cookies so we can understand how you use this website and make improvements.

[Accept analytics cookies](#)

[Reject analytics cookies](#)

[View cookies \(/cookies\)](#)



[About](#) [Documentation](#) [Get started](#) [Support](#)  
[Sign in](#)

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

[GOV.UK Services](#) > [GOV.UK One Login](#) > [Documentation](#) > [Design recommendations](#)

[Technical documentation \(/documentation\)](#)

[Sign in user journey maps \(/documentation/user-journeys\)](#)

[Proving identity journey maps \(/documentation/identity-journeys\)](#)

[GOV.UK One Login prototype \(/documentation/end-to-end-prototype/identity-journeys\)](#)

[Design recommendations \(/documentation/design-recommendations\)](#)

[Let users create a GOV.UK One Login to save progress \(/documentation/design-recommendations/save-progress\)](#)

**Let users navigate to their GOV.UK One Login and sign out easily**  
(/documentation/design-recommendations/let-users-navigate-sign-out)

## Let users navigate to their GOV.UK One Login and sign out easily

### What is it?

The GOV.UK One Login service header gives users an easy, consistent route from your service to their GOV.UK One Login and a way to sign out.

It's different to the GOV.UK header and service navigation components from the GOV.UK Design System.

When signed in, users can navigate to their GOV.UK One Login easily so they can access the services they use with it and also change their sign in details.

### When to use it

You should use this pattern if your service is using GOV.UK One Login.

### When not to use it

Show users where to change their GOV.UK One Login credentials  
(/documentation/design-recommendations/change-credentials)

Moving to GOV.UK One Login if your service already proves users identities  
(/documentation/design-recommendations/migrating-users)

Moving to GOV.UK One Login if you need to sign in your users  
(/documentation/design-recommendations/migrating-sign-in-pattern)

What to say about GOV.UK One Login on start pages  
(/documentation/design-recommendations/start-page)

How to talk about GOV.UK One Login  
(/documentation/design-recommendations/how-to-talk-about)

Prepare your users to move to GOV.UK One Login  
(/documentation/design-recommendations/prepare-to-move)

Business users and GOV.UK One Login  
(/documentation/design-recommendations/business-users)

User groups who may find it harder to prove their identity  
(/documentation/design-recommendations/barriers-to-proving-identity)

The GOV.UK One Login service header might not be right for your service if it does not use GOV.UK branding. For example, [Apply to become a registered social worker in England](https://www.socialworkengland.org.uk/registration/apply-for-registration/) (<https://www.socialworkengland.org.uk/registration/apply-for-registration/>).

## How it works

The GOV.UK One Login service header provides consistent navigation for users. It has 2 sections.

1. A top level black section that allows your users to:

- navigate from your service to their GOV.UK One Login
- sign out of both your service and their GOV.UK One Login

2. A service level grey section for your service name and navigation menu.

Our research shows that using the top black section of the header for the GOV.UK One Login menu and then displaying the service name and menu on the grey level below, clearly shows these are 2 different spaces.

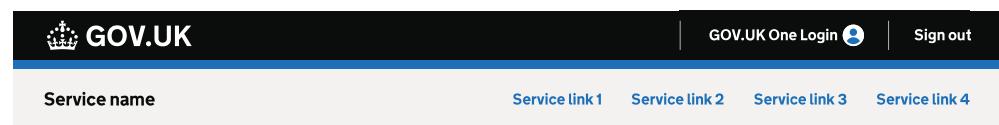
## Choose an option for the header

You can choose one of 3 options of the header to suit your service:

- Option 1: Default header
- Option 2: Header with no service menu
- Option 3: Header with a long service name or menu

### Option 1: Default header

Use this if your service name and menu links fit on one level.



### Desktop example

Service name

▼ Menu

## Mobile menus closed

GOV.UK One Login

Sign out

Service name

^ Close

Service link 1

Service link 2

## Mobile menus open

### Option 2: Header with no service menu

Use this if your service has no navigation menu.

GOV.UK

GOV.UK One Login

Sign out

Service name

### Desktop example

### Option 3: Header with a long service name or menu

Use this if your service name and menu links do not fit on one level.

GOV.UK

GOV.UK One Login

Sign out

Service that has a really long name that stretches quite far across

Service link 1 Service link 2 Service link 3 Service link 4 Service link 5 Service link 6 Service link 7

### Desktop example

## How to use the header

### Show the header on every page in your service

Use the header at the top of every page in your service when a user is signed in.

This gives your users a consistent way of signing out. It also helps increase their awareness of GOV.UK One Login.

Our research shows no evidence that the header distracts users in a service journey.

### Use specific menu labels

Avoid generic menu labels in your service navigation as users might think they link to their GOV.UK One Login.

For example, to link to your service homepage do not use labels such as ‘account’ or ‘home’ on their own. Instead, name the account or home it refers to such as ‘Your childcare account’ or ‘Dart charge home’. Or use a descriptive label. For example, if your service homepage gives users a list of applications they’ve submitted, you could label it ‘Your applications’ or ‘Dashboard’.

Our research shows that replacing generic service menu labels with service-specific ones helps users understand that the menu links go directly to pages within the service, not their GOV.UK One Login.

## How to link to your service homepage

Do not use the service name as a link to your service’s homepage. Instead, add it as the first link in the service menu, making sure you give it a specific label. This is better for accessibility and usability, as it gives users a clear indication of where the link is going to take them.

## Use complementary patterns when needed

Consider giving your users additional ways to change their sign in details, if your research or data suggests they are not sure where to go. This might be suitable if your service has a separate section for user details like email address and phone number.

## If you cannot use this header

You must give your users a way to navigate from your service to their GOV.UK One Login sign in details. See [additional ways or patterns to show users where to change their GOV.UK One Login credentials \(/documentation/design-recommendations/change-credentials\)](#).

You must provide a sign out link that signs users out of both their GOV.UK One Login and your service. [Read the technical documentation on signing your users out \(<https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/managing-your-users-sessions/#log-your-user-out-of-gov-uk-one-login>\)](#).

If you have any questions or need support, you can contact us using our [support form \(/support\)](#). We aim to reply within 2 working days.

## Help improve this header

If you would like to help us test the new header, or have any user insights to share, please get in touch through our [support form \(/contact-us\)](#) or [Slack channel \(https://ukgovernmentdigital.slack.com/archives/C02AQUJ6WTC\)](#).

## Get started

You can [get the code for the header and read the guidance on Github \(https://github.com/alphagov/di-govuk-one-login-service-header\)](#).

# Cookies on GOV.UK One Login

We'd like to use analytics cookies so we can understand how you use this website and make improvements.

[Accept analytics cookies](#)

[Reject analytics cookies](#)

[View cookies \(/cookies\)](#)



[About](#) [Documentation](#) [Get started](#) [Support](#)  
[Sign in](#)

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

[GOV.UK Services](#) > [GOV.UK One Login](#) > [Documentation](#)

[Technical documentation \(/documentation\)](#)

[Sign in user journey maps \(/documentation/user-journeys\)](#)

[Proving identity journey maps \(/documentation/identity-journeys\)](#)

[GOV.UK One Login prototype \(/documentation/end-to-end-prototype/identity-journeys\)](#)

[Design recommendations \(/documentation/design-recommendations\)](#)

[Let users create a GOV.UK One Login to save progress \(/documentation/design-recommendations/save-progress\)](#)

[Let users navigate to their GOV.UK One Login and sign out easily \(/documentation/design-recommendations/let-users-navigate-sign-out\)](#)

## Moving to GOV.UK One Login if you need to sign in your users

These design patterns will help your service migrate your users so they can sign in using GOV.UK One Login. Using these patterns means existing users will retain access to all of the information in their account.

These patterns work on the basis that ideally the same email address is used for both your service and GOV.UK One Login. You can find out more in our [technical documentation](#). (<https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/authenticate-your-user/#retrieve-user-information>) If this is not possible for the majority of your users, these patterns may not be right for your service. [Contact us \(/support\)](#) to talk about other possible approaches for your service.

These design patterns set out the steps GOV.UK One Login, or your service, should take. The design pattern you need to use depends on the specific circumstances of your users and service.

[Contact us \(/support\)](#) if:

Show users where to change their GOV.UK One Login credentials  
(/documentation/design-recommendations/change-credentials)

---

Moving to GOV.UK One Login if your service already proves users identities  
(/documentation/design-recommendations/migrating-users)

---

**Moving to GOV.UK One Login if you need to sign in your users**  
(/documentation/design-recommendations/migrating-sign-in-pattern)

---

What to say about GOV.UK One Login on start pages  
(/documentation/design-recommendations/start-page)

---

How to talk about GOV.UK One Login  
(/documentation/design-recommendations/how-to-talk-about)

---

Prepare your users to move to GOV.UK One Login  
(/documentation/design-recommendations/prepare-to-move)

---

Business users and GOV.UK One Login  
(/documentation/design-recommendations/business-users)

---

User groups who may find it harder to prove their identity  
(/documentation/design-recommendations/barriers-to-proving-identity)

---

- the following design patterns do not cover your circumstances
- you need the patterns in a different format (they're Figma files)
- you need more help

## Design pattern 1: when new users come to your service

We're working on making our documentation more accessible. If you have any problems accessing these Figma files please [contact us \(/support\)](#).

Pattern to use when new users come to your service  
(<https://www.figma.com/design/JkZaqVMn7YdlVIKaBZDLoC/GOV-UK-One-Login-Migration-Patterns?node-id=34-1225&t=PiyTimwTKPZNxe2K-4>)

1. Your service sends the user to GOV.UK One Login.
2. The user signs in or creates a GOV.UK One Login.
3. When the user has signed in to GOV.UK One Login, we'll send your service the user's unique Subject ID and email address. Then you can check if the user is new to your service. You may also wish to do other checks to confirm that they are a new user.
4. You'll use the unique Subject ID to connect ('bind') GOV.UK One Login access to your service for that user. Do not use the user's email address as the user can change the email they use with GOV.UK One Login.
5. If they are a new user, you may want to do eligibility checks before you provide access to your service.

## Design pattern 2: when existing users of your service access your service online

Pattern to use when existing users have already accessed your service online  
(<https://www.figma.com/design/JkZaqVMn7YdlVIKaBZDLoC/GOV-UK-One-Login-Migration-Patterns?node-id=34-1226&t=PiyTimwTKPZNxe2K-4>)

1. Your service sends the user to GOV.UK One Login.
2. The user signs in or creates a GOV.UK One Login.

3. The user will, ideally, have used the same email address for your service and GOV.UK One Login when they migrate to simplify the journey.
4. When the user has signed in to GOV.UK One Login, we'll send your service a unique identifier code and the email address. Then you can check if the user has an existing account with you.
5. You may want to do further checks to confirm this is an existing user who matches a record in your service. You could, for example, ask single or multiple security questions that only the user will be able to answer or get the user to sign in using their existing sign in credentials.
6. If you are happy with the match to a record in your service, then you can connect ('bind') them to your service using the unique Subject ID. Do not use their email address as that can change.
7. Your user will now be able to access your service and their account information using GOV.UK One Login.

## How to help users that you cannot match on email address

Example options for matching a user

(<https://www.figma.com/design/JkZaqVMn7YdIVIKaBZDLoC/GOV.UK-One-Login-Migration-Patterns?node-id=34-1227&t=PiyTimwTKPZNxe2K-4>)

If an existing user returns to your service, but has not used the same email address with their GOV.UK One Login, you can try to match them by asking:

- if they've used your service before – if they have, send a one time passcode to the email address they used with your service
- a security question(s) about their account that only they should know the answer(s) to
- them to sign in using their existing sign in details

## Design pattern 3: when users return to your service with a GOV.UK One Login

Pattern to use when users return to your service

(<https://www.figma.com/design/JkZaqVMn7YdIVIKaBZDLoC/GOV.UK-One-Login-Migration-Patterns?node-id=34-1228&t=PiyTimwTKPZNxe2K-4>)

1. Your service sends the user to GOV.UK One Login.
2. The user signs in to GOV.UK One Login.
3. When the user has signed in to GOV.UK One Login, we'll send your service the user's unique Subject ID. Then you can check if the user is a returning user.
4. If they are a returning user, you can provide access to your service.

## When to use this design pattern

Use this design pattern if all of the following circumstances apply:

- your service uses an email address as the user ID
- you want your users to retain access to all of the existing information they hold in your service
- you only need to sign in your users but do not need to prove their identity

## What to consider when using this design pattern

If using this pattern, you'll need to consider:

- when, how and where you tell your users they should use the same email with GOV.UK One Login – this could be handled by sending users an email to say they need to change their existing email address with you if it's different from the one they want to use, or already use, with GOV.UK One Login.
- [how you'll handle any exception cases](#) and bind accounts where the email addresses do not match
- if you want an additional layer of security when a user signs in to your service for the first time – for example by asking them about the information you hold about them
- how your current access model may affect this pattern – for example if you currently allow multiple users access to the same account, or one user to access multiple accounts, or if you allow users to share email addresses
- how you might stop users signing in to your service using their old sign in details

# Cookies on GOV.UK One Login

We'd like to use analytics cookies so we can understand how you use this website and make improvements.

[Accept analytics cookies](#)

[Reject analytics cookies](#)

[View cookies \(/cookies\)](#)



[About](#) [Documentation](#) [Get started](#) [Support](#)  
[Sign in](#)

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

[GOV.UK Services](#) > [GOV.UK One Login](#) > [Documentation](#)

[Technical documentation \(/documentation\)](#)

[Sign in user journey maps \(/documentation/user-journeys\)](#)

[Proving identity journey maps \(/documentation/identity-journeys\)](#)

[GOV.UK One Login prototype \(/documentation/end-to-end-prototype/identity-journeys\)](#)

[Design recommendations \(/documentation/design-recommendations\)](#)

[Let users create a GOV.UK One Login to save progress \(/documentation/design-recommendations/save-progress\)](#)

[Let users navigate to their GOV.UK One Login and sign out easily \(/documentation/design-recommendations/let-users-navigate-sign-out\)](#)

## Moving to GOV.UK One Login if your service already proves users identities

We've created this design pattern to help your service migrate (or move) your users to GOV.UK One Login to prove their identity. This pattern assumes your service already proves users' identities using an existing identity service.

All of these migrations are initiated by the user when they sign in to your service. GOV.UK One Login does not currently migrate users in bulk uploads.

These design patterns are diagrams that set out the steps GOV.UK One Login, or your service should take. The design pattern you need to use depends on the specific circumstances for your users or service.

[Contact us \(/support\)](#) if:

- the following design patterns do not cover your circumstances
- you need the patterns in a different format (they're Mural files)

Show users where to change their GOV.UK One Login credentials  
(/documentation/design-recommendations/change-credentials)

**Moving to GOV.UK One Login if your service already proves users identities**  
(/documentation/design-recommendations/migrating-users)

**Moving to GOV.UK One Login if you need to sign in your users**  
(/documentation/design-recommendations/migrating-sign-in-pattern)

**What to say about GOV.UK One Login on start pages**  
(/documentation/design-recommendations/start-page)

**How to talk about GOV.UK One Login**  
(/documentation/design-recommendations/how-to-talk-about)

**Prepare your users to move to GOV.UK One Login**  
(/documentation/design-recommendations/prepare-to-move)

**Business users and GOV.UK One Login**  
(/documentation/design-recommendations/business-users)

**User groups who may find it harder to prove their identity**  
(/documentation/design-recommendations/barriers-to-proving-identity)

- you need more help

## Design pattern 1: when new users come to your service

Pattern to use when new users come to your service  
(<https://www.figma.com/design/JkZaqVMn7YdlVIKaBZDLoC/GOV.UK-One-Login-Migration-Patterns?node-id=153-1336&t=sBCTxjktqodwi8J0-4>)

1. Your service sends the user to GOV.UK One Login.
2. The user signs in or creates a GOV.UK One Login. This is the authentication stage.
3. When the user has signed in to GOV.UK One Login, we'll send your service a unique identifier code so you can check if the user is new to your service, or is a returning user.
4. If they are a new user, they'll have to prove their identity using GOV.UK One Login.

## Design pattern 2: when existing users have already proved their identity with your service

Pattern to use when existing users have already proved their identity with your service  
(<https://www.figma.com/design/JkZaqVMn7YdlVIKaBZDLoC/GOV.UK-One-Login-Migration-Patterns?node-id=153-1337&t=sBCTxjktqodwi8J0-4>)

1. Your service sends the user to GOV.UK One Login.
2. The user signs in or creates a GOV.UK One Login. This is the authentication stage.
3. When the user has signed in to GOV.UK One Login, we'll send your service a unique identifier code so you can check if the user is new to your service, or is a returning user.
4. If the user is signing in to GOV.UK One Login for the first time, and they match a record in your service, then you can connect ('bind') them to your service. This means the user will not need to prove their identity again with GOV.UK One Login. You'll need to use the identity record they created with your legacy identity service provider.

# Design pattern 3: when users return to your service

## Pattern to use when users return to your service

(<https://www.figma.com/design/JkZaqVMn7YdIVIKaBZDLoC/GOV-UK-One-Login-Migration-Patterns?node-id=153-1338&t=FFWMunUGE3v7bXXf-4>)

1. When a user returns to your service, you'll need to check their proof of identity using either:
  - your legacy identity service
  - GOV.UK One Login
2. You'll need to decide when you want to retire your legacy identity service. When you do this your existing users will also need to sign in to GOV.UK One Login and use that to prove their identity. Once all your users are on GOV.UK One Login, you'll be able to stop using your legacy identity service.

## When to use this design pattern

Use this design pattern if all of the following circumstances apply:

- your service's existing users have already proved their identities
- the existing identity service provider holds these users' data, and you'd like to reuse the data so they do not need to prove their identity again with GOV.UK One Login
- for new users, you'll stop using your existing identity service provider and start using GOV.UK One Login to prove their identity

## What to consider when using this design pattern

This pattern will mean:

- existing users do not need to prove their identities again
- you can decide when you start allowing existing users to move to GOV.UK One Login for identity proving

- you'll need to keep records about your existing users who've already proved their identity until they've all moved over to GOV.UK One Login, which will mean continuing to run your legacy identity proving solution for some time
- you'll need the ability to build new functionality, for example to check if a user proved their identity with GOV.UK One Login or your existing identity service

# Cookies on GOV.UK One Login

We'd like to use analytics cookies so we can understand how you use this website and make improvements.

[Accept analytics cookies](#)

[Reject analytics cookies](#)

[View cookies \(/cookies\)](#)



[About](#) [Documentation](#) [Get started](#) [Support](#)  
[Sign in](#)

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

[GOV.UK Services](#) > [GOV.UK One Login](#) > [Documentation](#)

[Technical documentation \(/documentation\)](#)

[Sign in user journey maps \(/documentation/user-journeys\)](#)

[Proving identity journey maps \(/documentation/identity-journeys\)](#)

[GOV.UK One Login prototype \(/documentation/end-to-end-prototype/identity-journeys\)](#)

[Design recommendations \(/documentation/design-recommendations\)](#)

[Let users create a GOV.UK One Login to save progress \(/documentation/design-recommendations/save-progress\)](#)

[Let users navigate to their GOV.UK One Login and sign out easily \(/documentation/design-recommendations/let-users-navigate-sign-out\)](#)

## Prepare your users to move to GOV.UK One Login

Before your service starts using GOV.UK One Login, you may want to tell your users this is happening.

The wording you use on your start page depends on how you're using GOV.UK One Login, for instance authentication and identity or only authentication, and how you're currently handling sign in and identity proving.

### 1. Existing service account, migrating users to GOV.UK One Login

How you sign in to [Name of service] is changing

From [Date] you'll sign in using GOV.UK One Login. You'll be able to create a GOV.UK One Login if you do not already have one.

You should use the same email address to create your GOV.UK One Login that you use for your [Name of account]. This is so you keep the existing information in your account.

Show users where to change their GOV.UK One Login credentials  
(/documentation/design-recommendations/change-credentials)

---

Moving to GOV.UK One Login if your service already proves users identities  
(/documentation/design-recommendations/migrating-users)

---

Moving to GOV.UK One Login if you need to sign in your users  
(/documentation/design-recommendations/migrating-sign-in-pattern)

---

What to say about GOV.UK One Login on start pages  
(/documentation/design-recommendations/start-page)

---

How to talk about GOV.UK One Login  
(/documentation/design-recommendations/how-to-talk-about)

---

Prepare your users to move to GOV.UK One Login  
(/documentation/design-recommendations/prepare-to-move)

---

Business users and GOV.UK One Login  
(/documentation/design-recommendations/business-users)

---

User groups who may find it harder to prove their identity  
(/documentation/design-recommendations/barriers-to-proving-identity)

---

## 2. Existing service account, not migrating users

How you sign in to [Name of service] is changing

From [Date] you'll sign in using GOV.UK One Login. You'll be able to create a GOV.UK One Login if you do not already have one.

You can also use your GOV.UK One Login to [access other government services](https://www.gov.uk/using-your-gov-uk-one-login/services) (<https://www.gov.uk/using-your-gov-uk-one-login/services>).

Depending on the service, your users may also want to know what happened to their old account, or at least the information in it.

## 3. No sign in currently on service

How you access [Name of service] is changing

You'll need to sign in using GOV.UK One Login to use [Name of service].

You'll be able to create a GOV.UK One Login if you do not already have one.

You can also use your GOV.UK One Login to [access other government services](https://www.gov.uk/using-your-gov-uk-one-login/services) (<https://www.gov.uk/using-your-gov-uk-one-login/services>).

## 4. Existing service account, migrating users and suggesting users change their email address before going live

How you sign in to [Name of service] is changing

From [Date] you'll sign in using GOV.UK One Login. You'll be able to create a GOV.UK One Login if you do not already have one.

You should use the same email address to create your GOV.UK One Login that you use for your [Name of account]. This is so you keep the existing information in your account.

You can change your email address you use with [Name of account] by signing in to your account. You must do this before [Insert date].

## **5. Identity service (where a service wants to mention that users may need to prove their identity again)**

How you sign in to [Name of service] is changing

From [Date] you'll sign in using GOV.UK One Login. You'll be able to create a GOV.UK One Login if you do not already have one.

You should use the same email address to create your GOV.UK One Login that you use for your [Name of account]. This is so you keep the existing information in your account.

You may need to prove your identity with GOV.UK One Login. You'll be told if you need to do this when you sign in. This is to keep your details safe and usually involves using photo ID like a passport or driving licence.

## **6. Services with multiple ways to sign in**

Some services may offer multiple ways to sign in for a period. This may be because:

- different user groups, for example individuals or business, may use separate sign in options
- you want to phase in the introduction of GOV.UK One Login alongside your existing sign in solution

If that's the case, and all your users are accessing the service through the same start page, a general call to action may not apply to all of the possible use cases.

Instead, you may want to change how you talk about signing in on your start page to make it more generic, and not to talk about a specific solution.

For example:

You'll need to sign in to use this service. If you do not already have sign in details, you'll be able to create

them.

When a user signs in, and you know more about them, you can ensure they use the correct sign in option for them.

# Cookies on GOV.UK One Login

We'd like to use analytics cookies so we can understand how you use this website and make improvements.

[Accept analytics cookies](#)

[Reject analytics cookies](#)

[View cookies \(/cookies\)](#)



[About](#) [Documentation](#) [Get started](#) [Support](#)  
[Sign in](#)

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

[GOV.UK Services](#) > [GOV.UK One Login](#) > [Documentation](#) > [Design recommendations](#)

[Technical documentation \(/documentation\)](#)

[Sign in user journey maps \(/documentation/user-journeys\)](#)

[Proving identity journey maps \(/documentation/identity-journeys\)](#)

[GOV.UK One Login prototype \(/documentation/end-to-end-prototype/identity-journeys\)](#)

[Design recommendations \(/documentation/design-recommendations\)](#)

**Let users create a GOV.UK One Login to save progress**  
([/documentation/design-recommendations/save-progress](#))

[Let users navigate to their GOV.UK One Login and sign out easily](#)  
([/documentation/design-](#)

## Let users create a GOV.UK One Login to save progress

These designs show you how to let users complete part of your journey, then create a GOV.UK One Login so they can come back and finish it later. [See journey map option 2 \(/documentation/user-journeys\)](#) for more context.

### Contents

- [Save and complete later link](#)
- [Start or resume a report or application](#)
- [Task list page](#)
- [Sign out interruption screen](#)
- [Confirmation email](#)

### Save and complete later link

recommendations/let-users-navigate-sign-out)

Show users where to change their GOV.UK One Login credentials  
(/documentation/design-recommendations/change-credentials)

Moving to GOV.UK One Login if your service already proves users identities  
(/documentation/design-recommendations/migrating-users)

Moving to GOV.UK One Login if you need to sign in your users  
(/documentation/design-recommendations/migrating-sign-in-pattern)

What to say about GOV.UK One Login on start pages  
(/documentation/design-recommendations/start-page)

How to talk about GOV.UK One Login  
(/documentation/design-recommendations/how-to-talk-about)

Prepare your users to move to GOV.UK One Login  
(/documentation/design-recommendations/prepare-to-move)

Business users and GOV.UK One Login  
(/documentation/design-recommendations/business-users)

User groups who may find it harder to prove their identity  
(/documentation/design-recommendations/barriers-to-proving-identity)

The screenshot shows a service form from GOV.UK. At the top, there's a header with the GOV.UK logo and the text 'Tell DVLA you've sold, transferred or bought a vehicle'. Below this is a back navigation link '< Back'. The main section is titled 'Date you sold the vehicle' with a placeholder 'For example, 27 3 2008'. There are three input fields labeled 'Day', 'Month', and 'Year', each with a small input box. Below these fields is a green 'Continue' button. At the bottom left of the form area is a blue 'Save and complete later' link.

## What is it?

A ‘Save and complete later’ link is the first point of interaction for a user when they decide they want to save their report or application, and come back to finish it later. The link is placed under the primary action of a page.

## When to use it

Use the ‘Save and complete later’ link on all pages within a service form.

## How it works

The link will take users into the Sign In journey. From there users have the option of creating a GOV.UK One Login or signing in if they already have one.

# Start or resume a report or application

[Back](#)**What do you want to do?** Start a new report to DVLA

You can save your progress at any point. You'll need a GOV.UK One Login. If you don't have a GOV.UK One Login, you can create one.

 Resume a saved report

You'll need to sign in to your GOV.UK One Login.

[Continue](#)**What is it?**

A ‘What do you want to do?’ screen asks users if they want to start a new report or application, or resume one they’ve previously saved. It’s their point of access for resuming an application.

**When to use it**

This screen is shown directly after the service start page, after the user has selected ‘Start now’.

**How it works**

Selecting the first radio button “Start a new report to [service]” will take users to the first question of the service’s form.

Selecting the second radio button “Resume a saved report” will take users to a ‘sign in to your GOV.UK One Login to resume’ page. Users must sign in to view their progress with the saved report or application and then carry on from where they left off.

---

**Task list page**

Tell DVLA you've sold, transferred or bought a vehicle

## Check your answers

### Success

#### Your progress has been saved

You can see and complete your [report to DVLA] below or you can [sign out](#).

Last saved: 16:17:45 on 07-07-2021

#### [Report] incomplete

You have completed 1 out of 3 sections.

##### 1. Your details

[Are you a motor trader?](#)

**COMPLETED**

[Your full name](#)

**COMPLETED**

##### 2. Your vehicle

[What have you done with your vehicle?](#)

**COMPLETED**

[Date you sold the vehicle](#)

**COMPLETED**

[Did you sell the vehicle privately, or to a motor trader?](#)

**IN PROGRESS**

[Name of the motor trader you sold the vehicle to](#)

**NOT STARTED YET**

Vehicle registration number

**CANNOT START YET**

Check vehicle details

**CANNOT START YET**

V5C document reference number

**CANNOT START YET**

##### 3. Submit

[Submit \[report\]](#)

**CANNOT START YET**

Tell DVLA you've sold, transferred or bought a vehicle

## Check your answers

#### [Report] incomplete

You have completed 1 out of 3 sections.

##### 1. Your details

[Are you a motor trader?](#)

**COMPLETED**

[Your full name](#)

**COMPLETED**

##### 2. Your vehicle

[What have you done with your vehicle?](#)

**COMPLETED**

[Date you sold the vehicle](#)

**COMPLETED**

[Did you sell the vehicle privately, or to a motor trader?](#)

**IN PROGRESS**

[Name of the motor trader you sold the vehicle to](#)

**NOT STARTED YET**

Vehicle registration number

**CANNOT START YET**

Check vehicle details

**CANNOT START YET**

V5C document reference number

**CANNOT START YET**

##### 3. Submit

[Submit \[report\]](#)

**CANNOT START YET**

## What is it?

A task list page shows all the tasks that a user has to complete as part of the application or report, with indicators to show the status of each task.

On one version of this page, a ‘success’ panel appears at the top to show that a user has saved their progress. On the other version, there’s no panel.

## When to use it

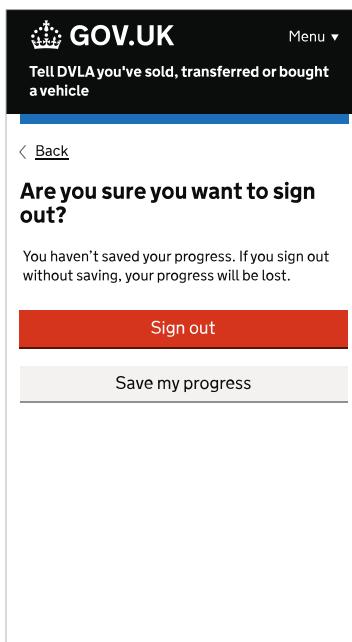
Use the version with the success panel immediately after a user has saved their progress on an application.

Use the version without the success panel after a user has signed in to resume an application.

## How it works

When a user selects a hyperlinked task in the task list, they can fill in their answers for that question (or change their answer if they'd previously completed it) and continue to fill in the form. If they save their progress again, they'll come back to the task list page.

## Sign out interruption screen



## What is it?

A sign out interruption screen warns signed-in users that their progress on a form will be lost if they do not save it before they sign out. They can choose to save their progress or sign out.

## When to use it

Use the sign out interruption screen when a signed-in user selects 'Sign out' without having saved their progress on

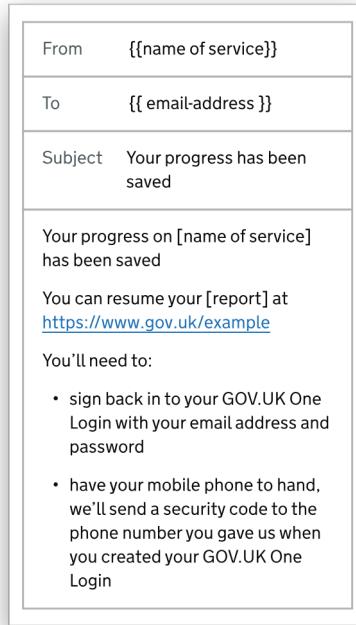
the form.

## How it works

The ‘Save my progress’ button will save the user’s progress on the form and send them to the task list page. The ‘Sign out’ button will sign them out of their GOV.UK One Login without saving their progress, and send them to the ‘You have signed out’ page.

---

## Confirmation email



## What is it?

A confirmation email is sent to the user to tell them that their progress on their application has been saved. It also gives them a route to resume their application.

## When to use it

Send the email to users after they have saved their progress on an application, report or other form.

## How it works

The content of the email tells users their report or application has been saved and how to resume it. The URL

takes the user to the ‘Sign in’ page.

# Cookies on GOV.UK One Login

We'd like to use analytics cookies so we can understand how you use this website and make improvements.

[Accept analytics cookies](#)

[Reject analytics cookies](#)

[View cookies \(/cookies\)](#)



[About](#) [Documentation](#) [Get started](#) [Support](#)  
[Sign in](#)

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

[GOV.UK Services](#) > [GOV.UK One Login](#) > [Documentation](#)

[Technical documentation \(/documentation\)](#)

[Sign in user journey maps \(/documentation/user-journeys\)](#)

[Proving identity journey maps \(/documentation/identity-journeys\)](#)

[GOV.UK One Login prototype \(/documentation/end-to-end-prototype/identity-journeys\)](#)

[Design recommendations \(/documentation/design-recommendations\)](#)

[Let users create a GOV.UK One Login to save progress \(/documentation/design-recommendations/save-progress\)](#)

[Let users navigate to their GOV.UK One Login and sign out easily \(/documentation/design-recommendations/let-users-navigate-sign-out\)](#)

## What to say about GOV.UK One Login on start pages

You'll need to update your service's start page to tell your users to use GOV.UK One Login to sign in to your service and, if your service needs them to, prove their identity.

### Moving your users to GOV.UK One Login

If your service already has a sign in or identity capability, you'll need to tell your users to create, or sign in, using GOV.UK One Login instead.

What you need to tell your users depends on how you decide to use GOV.UK One Login. For example, you may move:

- different groups of users at different times
- all of your users at the same time

Any messaging on your start page needs to work for all of your users as, until they sign in, you won't know which group they fall into.

Show users where to change their GOV.UK One Login credentials  
(/documentation/design-recommendations/change-credentials)

---

Moving to GOV.UK One Login if your service already proves users identities  
(/documentation/design-recommendations/migrating-users)

---

Moving to GOV.UK One Login if you need to sign in your users  
(/documentation/design-recommendations/migrating-sign-in-pattern)

---

**What to say about GOV.UK One Login on start pages**  
(/documentation/design-recommendations/start-page)

---

How to talk about GOV.UK One Login  
(/documentation/design-recommendations/how-to-talk-about)

---

Prepare your users to move to GOV.UK One Login  
(/documentation/design-recommendations/prepare-to-move)

---

Business users and GOV.UK One Login  
(/documentation/design-recommendations/business-users)

---

User groups who may find it harder to prove their identity  
(/documentation/design-recommendations/barriers-to-proving-identity)

---

## Content on start pages

Some users find it helpful to know that they'll either need their GOV.UK One login details, or that they'll need to create one as part of your service.

It can also help to reiterate what a user may need before they can create a GOV.UK One Login, for example an identity document.

However, bear in mind that:

- you can use other parts of the journey to explain what GOV.UK One Login is - for example within your service at the point where the user has to take action
- we've seen in research that a large proportion of users are drawn straight to the green start button
- there's often a lot of information on start pages competing for a user's attention
- you risk sending users off trying to find out what GOV.UK One Login is and whether they already have one or not

We recommend testing to see what approach works best for your users.

## Messaging throughout the journey

It can also be useful to include consistent and repeated messaging about GOV.UK One Login throughout the user journey.

We found that having the same information about GOV.UK One Login in multiple parts of the journey was helpful, for example on the create and sign in page or in a welcome email. For example, you could use the following content:

### About GOV.UK One Login

You need a GOV.UK One Login to access this service.

You can use your GOV.UK One Login to access some other government services.

## Start page patterns

### 1. Service with an existing account

You'll need a GOV.UK One Login to use this service.

If you haven't used GOV.UK One Login to access this service before, sign in using your existing sign in details. You'll then be asked to create or sign in to GOV.UK One Login.

In future, you should then always use your GOV.UK One Login to sign in to this service.

Start now button

## **2. Sign in ('authentication') service**

You need a GOV.UK One Login to sign in to this service. You can create one if you do not already have one.

Start now button

## **3. Identity service**

You'll need a GOV.UK One Login to use this service. You'll be able to create a GOV.UK One Login if you do not already have one.

You'll be told when you sign in if you need to prove your identity. This is to keep your details safe and normally involves using photo ID like a passport or driving licence.

Start now button

## **4. Identity service with an existing account, where existing users have already proved their identity**

You'll need a GOV.UK One Login to use this service. You'll be able to create a GOV.UK One Login if you do not already have one.

You'll be told when you sign in if you need to prove your identity. This is to keep your details safe and normally involves using photo ID like a passport or driving licence.

Start now button

## **Feedback**

We're still working on these patterns, so [please get in touch](#) ([/contact-us](#)) to let us know what works for your users.



# Cookies on GOV.UK One Login

We'd like to use analytics cookies so we can understand how you use this website and make improvements.

[Accept analytics cookies](#)

[Reject analytics cookies](#)

[View cookies \(/cookies\)](#)



[About](#) **Documentation** [Get started](#) [Support](#)  
[Sign in](#)

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

[GOV.UK Services](#) > [GOV.UK One Login](#) > [Documentation](#)

[Technical documentation \(/documentation\)](#)

[Sign in user journey maps \(/documentation/user-journeys\)](#)

[Proving identity journey maps \(/documentation/identity-journeys\)](#)

[GOV.UK One Login prototype \(/documentation/end-to-end-prototype/identity-journeys\)](#)

**Design recommendations**  
[\(/documentation/design-recommendations\)](#)

[Let users create a GOV.UK One Login to save progress \(/documentation/design-recommendations/save-progress\)](#)

[Let users navigate to their GOV.UK One Login and sign out easily \(/documentation/design-recommendations/let-users-navigate-sign-out\)](#)

## Design recommendations

We've created designs to help you incorporate GOV.UK One Login into your service.

The aim is to provide a consistent experience for users across all services that use GOV.UK One Login, and reduce duplication of effort by services.

However, you'll need to tailor them to your service. You do not have to use them at all if research shows something else works better for you.

## How we created our recommendations

The designs all use existing components from the [GOV.UK Design System \(<https://design-system.service.gov.uk>\)](#).

As this is a new type of journey, we've combined those components in new ways. We'll be sharing our findings with the GOV.UK Design System team.

The designs have all been tested with users. Contact us if you'd like more details on the research.

Show users where to change their GOV.UK One Login credentials  
(/documentation/design-recommendations/change-credentials)

---

Moving to GOV.UK One Login if your service already proves users identities  
(/documentation/design-recommendations/migrating-users)

---

Moving to GOV.UK One Login if you need to sign in your users  
(/documentation/design-recommendations/migrating-sign-in-pattern)

---

What to say about GOV.UK One Login on start pages  
(/documentation/design-recommendations/start-page)

---

How to talk about GOV.UK One Login  
(/documentation/design-recommendations/how-to-talk-about)

---

Prepare your users to move to GOV.UK One Login  
(/documentation/design-recommendations/prepare-to-move)

---

Business users and GOV.UK One Login  
(/documentation/design-recommendations/business-users)

---

User groups who may find it harder to prove their identity  
(/documentation/design-recommendations/barriers-to-proving-identity)

---

## Help improve our recommendations

Our design recommendations have all been tested with users, but we would love to know how they work in your service.

If you have user insights that would help us improve our recommendations or develop new ones, you can share them through our [support form](https://www.sign-in.service.gov.uk/contact-us) (<https://www.sign-in.service.gov.uk/contact-us>) or [Slack channel](https://ukgovernmentdigital.slack.com/archives/C02AQUJ6WTC) (<https://ukgovernmentdigital.slack.com/archives/C02AQUJ6WTC>).

# Cookies on GOV.UK One Login

We'd like to use analytics cookies so we can understand how you use this website and make improvements.

[Accept analytics cookies](#)

[Reject analytics cookies](#)

[View cookies \(/cookies\)](#)



One Login Beta

[About](#) [Documentation](#) [Get started](#) [Support](#) [Sign in](#)

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

## Email address sent

### What happens next

We'll now check your email address to make sure we can share the prototype with you.

We'll be in touch within a day.

[Go back to documentation](#)

# Cookies on GOV.UK One Login

We'd like to use analytics cookies so we can understand how you use this website and make improvements.

[Accept analytics cookies](#)

[Reject analytics cookies](#)

[View cookies \(/cookies\)](#)



One Login Beta

About Documentation Get started Support Sign in

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

< [Back](#)

## Enter your work email address

This should be an email address of a UK public body, for example a government department or organisation, or an arm's length body.

[Continue](#)

[Cancel \(/documentation/end-to-end-prototype/identity-journeys\)](#)

# Cookies on GOV.UK One Login

We'd like to use analytics cookies so we can understand how you use this website and make improvements.

[Accept analytics cookies](#)

[Reject analytics cookies](#)

[View cookies \(/cookies\)](#)



[About](#) [Documentation](#) [Get started](#) [Support](#)  
[Sign in](#)

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

[GOV.UK Services](#) > [GOV.UK One Login](#) > [Documentation](#)

[Technical documentation \(/documentation\)](#)

[Sign in user journey maps \(/documentation/user-journeys\)](#)

[Proving identity journey maps \(/documentation/identity-journeys\)](#)

**GOV.UK One Login prototype**  
(/documentation/end-to-end-prototype/identity-journeys)

[Design recommendations \(/documentation/design-recommendations\)](#)

[Let users create a GOV.UK One Login to save progress \(/documentation/design-recommendations/save-progress\)](#)

[Let users navigate to their GOV.UK One Login and sign out easily \(/documentation/design-](#)

## GOV.UK One Login prototype

We've created an HTML prototype for government service teams to use to explore journeys within GOV.UK One Login.

You'll need to have an email address from a UK public body to get access. If you do not have one, email [govuk-one-login@digital.cabinet-office.gov.uk](mailto:govuk-one-login@digital.cabinet-office.gov.uk), including who you work for and why you want access to the prototype.

We'll update the prototype regularly to reflect live journeys, but it's not an exact copy. It's also not production code.

[Get access](#)

recommendations/let-users-navigate-sign-out)

---

Show users where to change their GOV.UK One Login credentials  
(/documentation/design-recommendations/change-credentials)

---

Moving to GOV.UK One Login if your service already proves users identities  
(/documentation/design-recommendations/migrating-users)

---

Moving to GOV.UK One Login if you need to sign in your users  
(/documentation/design-recommendations/migrating-sign-in-pattern)

---

What to say about GOV.UK One Login on start pages  
(/documentation/design-recommendations/start-page)

---

How to talk about GOV.UK One Login  
(/documentation/design-recommendations/how-to-talk-about)

---

Prepare your users to move to GOV.UK One Login  
(/documentation/design-recommendations/prepare-to-move)

---

Business users and GOV.UK One Login  
(/documentation/design-recommendations/business-users)

---

User groups who may find it harder to prove their identity  
(/documentation/design-recommendations/barriers-to-proving-identity)

---



# Cookies on GOV.UK One Login

We'd like to use analytics cookies so we can understand how you use this website and make improvements.

[Accept analytics cookies](#)

[Reject analytics cookies](#)

[View cookies \(/cookies\)](#)



[About](#) [Documentation](#) [Get started](#) [Support](#)  
[Sign in](#)

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

[GOV.UK Services](#) > [GOV.UK One Login](#) > [Documentation](#)

[Technical documentation \(/documentation\)](#)

[Sign in user journey maps \(/documentation/user-journeys\)](#)

[Proving identity journey maps \(/documentation/identity-journeys\)](#)

[GOV.UK One Login prototype \(/documentation/end-to-end-prototype/identity-journeys\)](#)

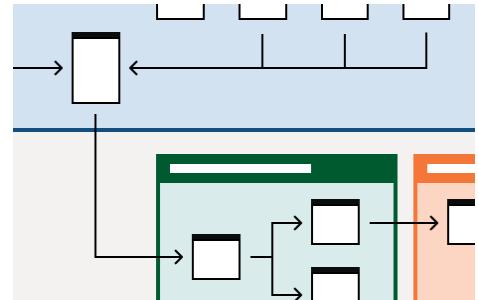
[Design recommendations \(/documentation/design-recommendations\)](#)

[Let users create a GOV.UK One Login to save progress \(/documentation/design-recommendations/save-progress\)](#)

[Let users navigate to their GOV.UK One Login and sign out easily \(/documentation/design-](#)

## Proving identity journey maps

We have detailed journey process maps which explain the end to end process of how your users flow through the entire GOV.UK One Login product.



The process maps are split into all the different journey stages.

They are designed to give you an overview of the end to end GOV.UK One Login journey.

[Go to proving identity journey maps \(opens in a new tab\)](#)  
(<https://www.figma.com/designs/6D6nLrW4MayhrlaJ4N7FHm/GOV.UK-One-Login-Identity-checking-user->

recommendations/let-users-navigate-sign-out)

---

Show users where to change their GOV.UK One Login credentials  
(/documentation/design-recommendations/change-credentials)

---

Moving to GOV.UK One Login if your service already proves users identities  
(/documentation/design-recommendations/migrating-users)

---

Moving to GOV.UK One Login if you need to sign in your users  
(/documentation/design-recommendations/migrating-sign-in-pattern)

---

What to say about GOV.UK One Login on start pages  
(/documentation/design-recommendations/start-page)

---

How to talk about GOV.UK One Login  
(/documentation/design-recommendations/how-to-talk-about)

---

Prepare your users to move to GOV.UK One Login  
(/documentation/design-recommendations/prepare-to-move)

---

Business users and GOV.UK One Login  
(/documentation/design-recommendations/business-users)

---

User groups who may find it harder to prove their identity  
(/documentation/design-recommendations/barriers-to-proving-identity)

---

[journey-maps?node-id=2001-814&t=XgOS7raJsReIpWWw-4\)](#)



# Cookies on GOV.UK One Login

We'd like to use analytics cookies so we can understand how you use this website and make improvements.

[Accept analytics cookies](#)

[Reject analytics cookies](#)

[View cookies \(/cookies\)](#)



[About](#) [Documentation](#) [Get started](#) [Support](#)  
[Sign in](#)

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

[GOV.UK Services](#) > [GOV.UK One Login](#) > [Documentation](#) > [Sign in user journey maps](#)

[Technical documentation \(/documentation\)](#)

**Sign in user journey maps**  
[\(/documentation/user-journeys\)](#)

[Proving identity journey maps](#)  
[\(/documentation/identity-journeys\)](#)

[GOV.UK One Login prototype](#)  
[\(/documentation/end-to-end-prototype/identity-journeys\)](#)

[Design recommendations](#)  
[\(/documentation/design-recommendations\)](#)

[Let users create a GOV.UK One Login to save progress](#)  
[\(/documentation/design-recommendations/save-progress\)](#)

[Let users navigate to their GOV.UK One Login and sign out easily](#)  
[\(/documentation/design-recommendations/let-users-navigate-sign-out\)](#)

## Sign in user journey maps

There are 2 different ways of using GOV.UK One Login in your service. You can see a user journey map for both options. For a more detailed view, request access to the [GOV.UK One Login prototype \(<https://www.sign-in.service.gov.uk/documentation/end-to-end-prototype/identity-journeys>\)](#).

Both options have been through multiple rounds of user testing.

We're working on making our documentation more accessible. If you have any problems accessing these journey maps, please [contact us \(/contact-us\)](#).

### Option 1: users create a GOV.UK One Login upfront

This journey map shows a user creating a GOV.UK One Login at the start of your service journey.

Show users where to change their GOV.UK One Login credentials  
(/documentation/design-recommendations/change-credentials)

Moving to GOV.UK One Login if your service already proves users identities  
(/documentation/design-recommendations/migrating-users)

Moving to GOV.UK One Login if you need to sign in your users  
(/documentation/design-recommendations/migrating-sign-in-pattern)

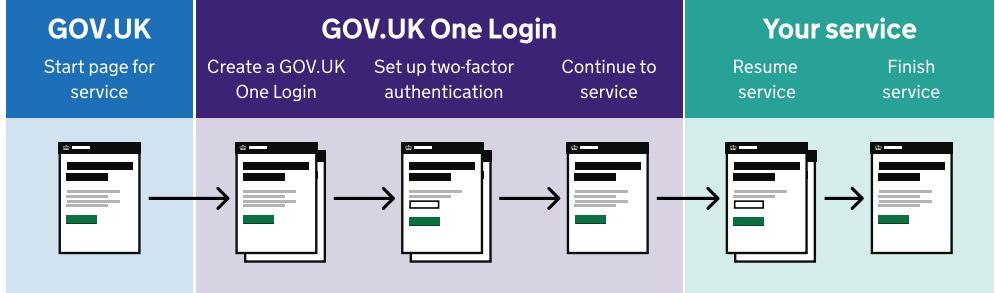
What to say about GOV.UK One Login on start pages  
(/documentation/design-recommendations/start-page)

How to talk about GOV.UK One Login  
(/documentation/design-recommendations/how-to-talk-about)

Prepare your users to move to GOV.UK One Login  
(/documentation/design-recommendations/prepare-to-move)

Business users and GOV.UK One Login  
(/documentation/design-recommendations/business-users)

User groups who may find it harder to prove their identity  
(/documentation/design-recommendations/barriers-to-proving-identity)



Users start at your service's start page, usually on GOV.UK. They create a GOV.UK One Login using their email address, a password and two-factor authentication. Then, they're sent to your service to complete their journey.

This option is usually best for your service if you need all your users to create a GOV.UK One Login before they can do anything.

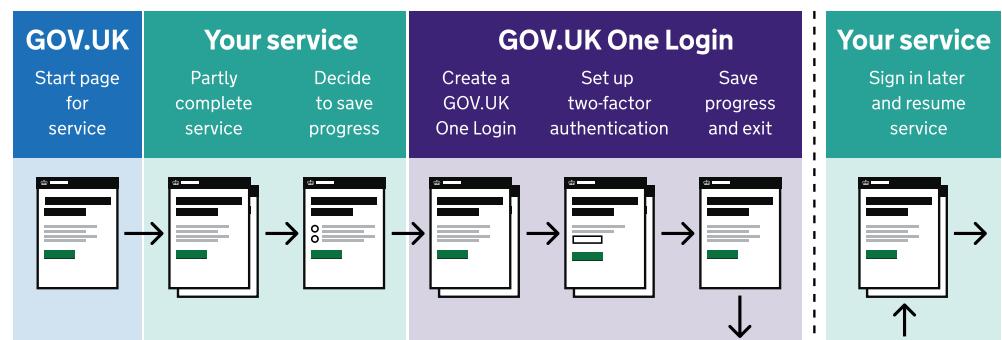
### See a detailed map of the user-facing screens

[View option 1 in Figma \(opens in new tab\)](#)  
(<https://www.figma.com/design/rAC00C9Lq0cNJ2VKC5jwy5/GOV-UK-One-Login-sign-in-user-journey-maps?node-id=1-49>)

Best if you want to copy the screens into your own journey maps and play around with the designs.

## Option 2: users create a GOV.UK One Login to save progress

This journey map shows a user creating a GOV.UK One Login in the middle of your service journey.



Users start at your service's start page, usually on GOV.UK. They complete part of your journey, then decide they want to finish it later.

To save their progress, they create a GOV.UK One Login using their email address, a password and two-factor authentication.

Later, they return to your service and resume their journey by signing in to their GOV.UK One Login.

This option is usually best for your service if you do not need all your users to create a GOV.UK One Login, but want to give them the option of saving their progress.

## See a detailed map of the user-facing screens

[View option 2 in Figma \(opens in new tab\)](#)  
(<https://www.figma.com/design/raco0c9lq0cnj2vkc5jwy5/GOV-UK-One-Login-sign-in-user-journey-maps?node-id=0-1>)

Best if you want to copy the screens into your own journey maps and play around with the designs.

## View design recommendations

See [Let users create a GOV.UK One Login to save progress](#) (<https://www.sign-in.service.gov.uk/documentation/design-recommendations/save-progress>) for more detail on how to incorporate the ‘save and complete later’ journey into your service.

# Cookies on GOV.UK One Login

We'd like to use analytics cookies so we can understand how you use this website and make improvements.

[Accept analytics cookies](#)

[Reject analytics cookies](#)

[View cookies \(/cookies\)](#)



[About](#) [Documentation](#) [Get started](#) [Support](#)  
[Sign in](#)

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

[GOV.UK Services](#) > [GOV.UK One Login](#) > [Documentation](#)

[Technical documentation \(/documentation\)](#)

[Sign in user journey maps \(/documentation/user-journeys\)](#)

[Proving identity journey maps \(/documentation/identity-journeys\)](#)

[GOV.UK One Login prototype \(/documentation/end-to-end-prototype/identity-journeys\)](#)

[Design recommendations \(/documentation/design-recommendations\)](#)

[Let users create a GOV.UK One Login to save progress \(/documentation/design-recommendations/save-progress\)](#)

[Let users navigate to their GOV.UK One Login and sign out easily \(/documentation/design-recommendations/let-users-navigate-sign-out\)](#)

## Technical documentation

Our technical documentation will help you:

- understand how GOV.UK One Login works
- test GOV.UK One Login in our integration environment

The documentation covers both the authentication and identity checking parts of GOV.UK One Login.

[Read our technical documentation \(opens in new tab\) \(<https://docs.sign-in.service.gov.uk>\)](#)

Show users where to  
change their GOV.UK One  
Login credentials  
(/documentation/design-  
recommendations/change-  
credentials)

---

Moving to GOV.UK One  
Login if your service already  
proves users identities  
(/documentation/design-  
recommendations/migrating-  
users)

---

Moving to GOV.UK One  
Login if you need to sign in  
your users  
(/documentation/design-  
recommendations/migrating-  
sign-in-pattern)

---

What to say about GOV.UK  
One Login on start pages  
(/documentation/design-  
recommendations/start-page)

---

How to talk about GOV.UK  
One Login  
(/documentation/design-  
recommendations/how-to-talk-  
about)

---

Prepare your users to move  
to GOV.UK One Login  
(/documentation/design-  
recommendations/prepare-to-  
move)

---

Business users and  
GOV.UK One Login  
(/documentation/design-  
recommendations/business-  
users)

---

User groups who may find it  
harder to prove their identity  
(/documentation/design-  
recommendations/barriers-to-  
proving-identity)

---

# Cookies on GOV.UK One Login

We'd like to use analytics cookies so we can understand how you use this website and make improvements.

[Accept analytics cookies](#)

[Reject analytics cookies](#)

[View cookies \(/cookies\)](#)



One Login Beta

About Documentation **Get started** Support Sign in

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

[GOV.UK Services](#) > [GOV.UK One Login](#) > [Get started](#) > [Public beta](#)

## Find out more about our public beta

GOV.UK One Login is currently in public beta. This means we're making it available to any eligible government service that needs to use it.

### Who can use GOV.UK One Login

GOV.UK One Login is available to services:

- run by central government departments
- in beta or live
- for the general public or specialist groups, for example teachers
- that need to sign in their users or check their identities
- that can integrate with GOV.UK One Login - this means you need access to developers who are comfortable working with the OpenID Connect (OIDC) protocol and the processes set out in our [technical documentation](#) (<https://docs.sign-in.service.gov.uk/integrate-with-integration-environment>)

GOV.UK One Login is not currently available to the NHS, police or local authorities.

If you're not sure if you can use GOV.UK One Login, you can [contact the team \(/support\)](#).

## What to expect

If you join the beta, you'll:

- get access to GOV.UK One Login for all your live traffic
- be able to sign in your users and check their identities if you need to
- get support from our team throughout

We'll need you to:

- integrate with GOV.UK One Login to allow your users to sign in to your service and check their identities if you need to
- give us regular feedback about your experiences of using and onboarding to GOV.UK One Login - we'll agree on how and how often you give feedback to make sure it works for both of us

## The joining process

If you meet our joining criteria and would like to take part, [register your interest \(/register\)](#).

You can then speak to us about your service and access our integration environment to see how GOV.UK One Login works with your service. If you decide GOV.UK One Login meets your needs, email us to confirm you'd like to join the beta.

We'll guide you through the process of getting your service live with GOV.UK One Login, agree a go live date and tell you what we need from you. For example, we need to know about your fraud approach and user support set-up.

You'll also need to sign our Memorandum of Understanding (MoU) before you can go live with GOV.UK One Login.

If you have any questions, [contact the team \(/support\)](#).

# Cookies on GOV.UK One Login

We'd like to use analytics cookies so we can understand how you use this website and make improvements.

[Accept analytics cookies](#)

[Reject analytics cookies](#)

[View cookies \(/cookies\)](#)



One Login Beta

About Documentation **Get started** Support Sign in

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

[GOV.UK Services](#) > [GOV.UK One Login](#) > [Get started](#)

## Get started with GOV.UK One Login

You can use GOV.UK One Login if you're from a central government service and need to sign in your users or check their identities.

[Read more about GOV.UK One Login \(/about\)](#)

## Get in touch or try GOV.UK One Login

Register your interest if you think GOV.UK One Login is right for your service. We'll contact you within 5 days to find out more about what your service needs.

[Register your interest](#)

## Try GOV.UK One Login

Set up an admin tool account to try GOV.UK One Login in our integration environment. You'll need a:

- government email address

- mobile phone number

Create admin tool account

---

or [sign in](https://admin.sign-in.service.gov.uk/sign-in) (<https://admin.sign-in.service.gov.uk/sign-in>)

## Contact us

Use our [online form \(/support\)](#) to report problems, ask questions or suggest improvements.

You can also get in touch using our [Slack channel](#) (<https://ukgovernmentdigital.slack.com/archives/C02AQUJ6WTC>)

Also, find out more about [GOV.UK Wallet](#) (<https://www.gov.uk/guidance/using-govuk-wallet-in-government>) or email [gov.uk.wallet-queries@digital.cabinet-office.gov.uk](mailto:gov.uk.wallet-queries@digital.cabinet-office.gov.uk) if you're interested in using it.

# Cookies on GOV.UK One Login

We'd like to use analytics cookies so we can understand how you use this website and make improvements.

[Accept analytics cookies](#)

[Reject analytics cookies](#)

[View cookies \(/cookies\)](#)



One Login Beta

About Documentation Get started Support Sign in

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

## Sorry, the service is unavailable

You will be able to use the service later.

If you need help urgently, contact us using our [support form](#) (<https://www.sign-in.service.gov.uk/contact-us>) or via our [Slack channel](#) (<https://ukgovernmentdigital.slack.com/archives/C02AQUJ6WTC>).

# Cookies on GOV.UK One Login

We'd like to use analytics cookies so we can understand how you use this website and make improvements.

[Accept analytics cookies](#)

[Reject analytics cookies](#)

[View cookies \(/cookies\)](#)



About Documentation Get started Support Sign in

Beta This is a new service – your [feedback \(opens in a new tab\) \(/contact-us\)](#) will help us to improve it.

GOV.UK Services

[GOV.UK One Login](#)

## Let users sign in and prove their identities to use your service

Try GOV.UK One Login if you work on a central government service.



[Get started >](#)

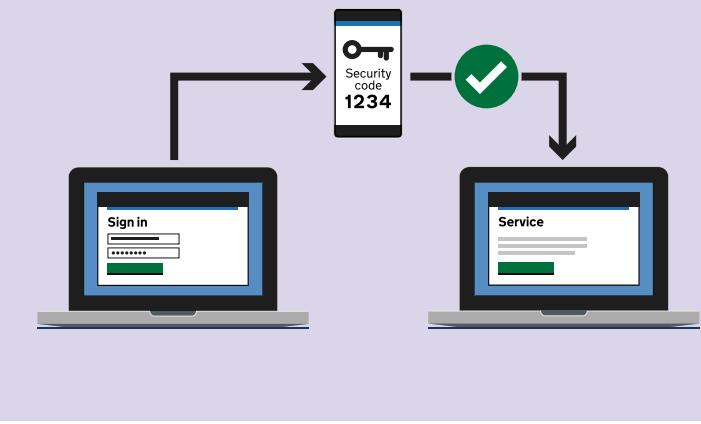
# Let users sign in

Let your users sign in to your service with their email address, password and two-factor authentication.

They can use these details to access all services that use GOV.UK One Login. They can also see the services they've used in one place.

[Read more about sign in and authentication \(/about/signing-users-in\)](#)

[Explore sign in and authentication journey maps \(/documentation/user-journeys\)](#)



## Check the identity of your users

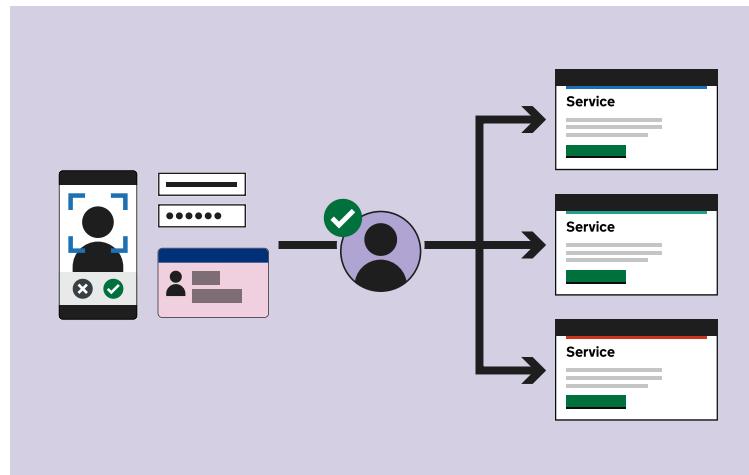
Get your users' identity checks done centrally by using GOV.UK One Login.

You'll get confirmation that the person is who they say they are without having to do any checks yourself.

Users will be able to reuse these checks to access other government services.

[Read more about identity checking \(/about/checking-users-identities\)](#)

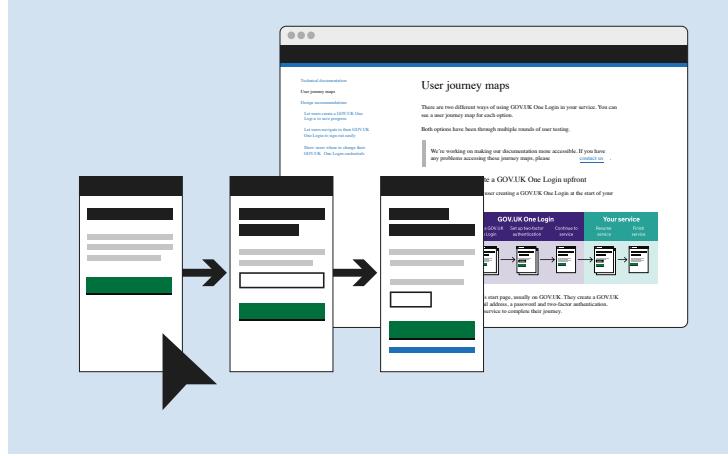
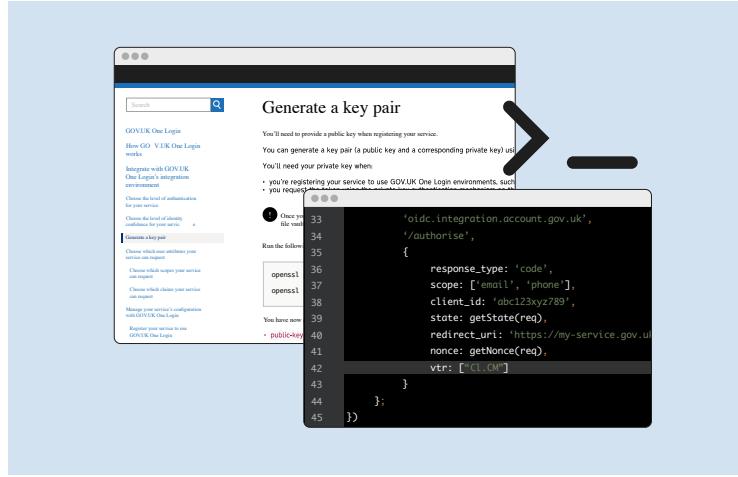
[Explore identity checking journey maps \(/documentation/identity-journeys\)](#)



## Get set up quickly and easily

See how GOV.UK One Login works with your service in our integration environment. Integrate at your own pace by following our [technical documentation \(opens in new tab\)](#) (<https://docs.sign-in.service.gov.uk>).

Incorporate our researched, accessible [user journeys \(/documentation/user-journeys\)](#) into your existing service journey.



# Try GOV.UK One Login

You can try out GOV.UK One Login if you're a central government service and need to sign in your users or check their identities.

# Get started

## Join our mailing list

If you have a government or public sector email address, [join our mailing list \(/mailing-list\)](#) to stay up to date with our progress.

You'll get updates on our work and invitations to our regular cross-government show and tells. You'll also get invited to take part in user research.

[Table of contents](#)

# Page not found

**If you typed the web address, check it is correct.**

If you pasted the web address, check you copied the entire address.

You can contact us on our [GOV.UK One Login Slack channel](#) (<https://ukgovernmentdigital.slack.com/archives/C02AQUJ6WTC>) if you need help.

This page was last reviewed on 19 December 2023.

[View source](#) [Report problem](#) [GitHub Repo](#)



[Accessibility](#)

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Accessibility

## Accessibility statement for GOV.UK One Login technical documentation

This accessibility statement applies to the GOV.UK One Login technical documentation at <https://docs.sign-in.service.gov.uk/>.

This website is run by the GOV.UK One Login team at the Government Digital Service (GDS). We want as many people as possible to be able to use this website. For example, that means you should be able to:

- change colours, contrast levels and fonts
- zoom in up to 300% without problems
- navigate most of the website using just a keyboard
- navigate most of the website using speech recognition software
- listen to most of the website using a screen reader (including the most recent versions of JAWS, NVDA and VoiceOver)

We've also made the website text as simple as possible to understand.

[AbilityNet](https://abilitynet.org.uk/) (<https://abilitynet.org.uk/>) has advice on making your device easier to use if you have a disability.

### How accessible this website is

This website is fully compliant with the Web Content Accessibility Guidelines version 2.1 AA standard.

### What to do if you cannot access parts of this website

If you need information on this website in a different format like accessible PDF, large print, easy read, audio recording or braille, [use the Support page to contact the GOV.UK One Login team](/support/#support) (</support/#support>) with details of your request.

We'll aim to reply in 3 working days.

## Reporting accessibility problems with this website

We're always looking to improve the accessibility of this website. If you find any problems not listed on this page or think we're not meeting accessibility requirements, [use the Support page to contact the GOV.UK One Login team \(/support/#support\)](#).

## Enforcement procedure

The Equality and Human Rights Commission (EHRC) is responsible for enforcing the Public Sector Bodies (Websites and Mobile Applications) (No. 2) Accessibility Regulations 2018 (the 'accessibility regulations'). If you're not happy with how we respond to your complaint, [contact the Equality Advisory and Support Service \(EASS\) \(https://www.equalityadvisoryservice.com/\)](#).

## Technical information about this website's accessibility

GDS is committed to making its website accessible, in accordance with the Public Sector Bodies (Websites and Mobile Applications) (No. 2) Accessibility Regulations 2018.

## Compliance status

This website is fully compliant with the Web Content Accessibility Guidelines version 2.1 AA standard.

## Preparation of this accessibility statement

This statement was prepared on 07 October 2021.

This website was last tested in October 2021.

This page was last reviewed on 7 October 2021.



## Accessibility

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Choose the level of authentication for your service

You'll need to choose the level of authentication your service will require your users to have. You can find help on selecting an appropriate level of protection in the [guidance on using authenticators to protect an online service, also known as 'GPG 44'](https://www.gov.uk/government/publications/authentication-credentials-for-online-government-services/giving-users-access-to-online-services#choosing-an-authenticator) (<https://www.gov.uk/government/publications/authentication-credentials-for-online-government-services/giving-users-access-to-online-services#choosing-an-authenticator>).

GOV.UK One Login uses '[Vectors of Trust](https://datatracker.ietf.org/doc/html/rfc8485)' (<https://datatracker.ietf.org/doc/html/rfc8485>). Your service can use these Vectors of Trust to request the right level of authentication for your users to gain access to your service. You'll include your vector in the query string as part of the request you make when you integrate with Authorization Code Flow.

GOV.UK One Login currently supports the following authentication levels, also known as 'levels of protection' in GPG 44.

Levels of protection	Vector value	Description of the levels of protection
Low level of protection	<code>C1</code> (credential low)	<p>This vector requires your users to have a username and password combination.</p> <p>You should only use this option if your service does not hold personal information about your users, for example if your service is about booking in an MOT.</p> <p>All services use <code>C1.Cm</code> as the authentication level by default, unless you change your authentication level to <code>C1</code>.</p> <p>If you request <code>C1</code>, you will not be able to request identity attributes.</p>
Medium level of protection	<code>C1.Cm</code> (credential medium)	<p>This vector requires your users to have a username and password combination, as well as using two-factor authentication (2FA). GOV.UK One Login currently supports 2FA either through a one-time password sent through SMS, or an authenticator app.</p> <p>All services use <code>C1.Cm</code> as the authentication level by</p>

default, unless you change your authentication level to [C1](#).

If you need to request identity attributes, you must request [C1.Cm](#).

---

You'll include your level of authentication in your request to the [/authorize](#) endpoint.

Once you have chosen your level of authentication, you'll need to [choose the level of identity confidence](#) ([/before-integrating/choose-the-level-of-identity-confidence/](#)) if your service needs identity proving.

If your service does not need identity proving, you can move on to [generate a key pair](#) ([/before-integrating/set-up-your-public-and-private-keys/#create-a-key-pair](#)).

This page was last reviewed on 11 November 2022.

[View source](#) [Report problem](#) [GitHub Repo](#)

---



[Accessibility](#)

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Choose the level of identity confidence for your service

Using identity proving functionality is optional. If your service needs identity proving, you'll need to choose the level of identity confidence your service needs.

You may need different levels of identity confidence at different points in your user journey. You can set the level of identity confidence your service needs for each request you make to GOV.UK One Login. Find out when and why to check someone's identity in the [guidance about how to prove and verify someone's identity, also known as 'GPG 45'](#) (<https://www.gov.uk/government/publications/identity-proofing-and-verification-of-an-individual/how-to-prove-and-verify-someones-identity>).

GOV.UK One Login uses '[Vectors of Trust](https://datatracker.ietf.org/doc/html/rfc8485)' (<https://datatracker.ietf.org/doc/html/rfc8485>). Your service can use Vectors of Trust to request the right level of identity confidence for your users to gain access to the relevant parts of your service. You'll include your vector in the query string as part of the request to the `/authorize` endpoint you make when you integrate with Authorization Code Flow.

Levels of identity confidence	Vector value	Description of the levels of identity confidence
No identity confidence	P0	By default, GOV.UK One Login will not return a level of identity confidence.
Low identity confidence	P1	A basic level of identity confidence, which reduces your service's risk of accepting impostors or fake identities with fabricated credentials, otherwise known as 'synthetic identities'.
Medium identity confidence	P2	A higher level of identity confidence to further reduce your service's risk of accepting impostors or fake identities with fabricated credentials, otherwise known as 'synthetic identities'.  To request a medium level of identity confidence ( P2 ), you must have specified <code>C1.Cm</code> (the medium level of

authentication) when you chose the level of authentication for your service.

---

Now you've chosen your level of identity confidence, you can [generate a key pair \(/before-integrating/set-up-your-public-and-private-keys/#create-a-key-pair\)](#).

This page was last reviewed on 11 November 2022.

[View source](#) [Report problem](#) [GitHub Repo](#)

---



## Accessibility

## **OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Choose which user attributes your service can request

Your service can request certain user attributes. To do this, you need to choose which ‘scopes’ and ‘claims’ your service will use and include these when you make your request to the `/authorize` endpoint.

OpenID Connect (OIDC) scopes are identifiers your application uses during authentication to authorise access to a user’s attributes, such as an email address. Each scope returns a set of user attributes contained within it. OIDC calls this set of user attributes ‘claims’.

The user attributes and how you request them will depend on whether you are requesting authentication only, or authentication with a level of identity confidence.

Type of request you’re making	What type of user attributes you can request
-------------------------------	--

Authentication only	You can only <a href="#">request user attributes using scopes (/before-integrating/choose-which-user-attributes-your-service-can-request/#choose-which-scopes-your-service-can-request)</a> .
---------------------	---

Authentication and identity proving	You can request user attributes using a combination of scopes and claims, depending on what your service needs.
-------------------------------------	---

You’ll need to agree which scopes and claims you want to use when you [register your service to use GOV.UK One Login \(/before-integrating/register-and-manage-your-service/\)](#).

## Choose which scopes your service can request

`openid` is the only scope you must include. You can choose to include other scopes for your request to the `/authorize` endpoint depending on the user attributes your service needs.

You can find details of the scopes in the following table.

Scope	Required or optional	Description
openid	Required	<p>OIDC requests to the <code>/authorize</code> endpoint must contain the <code>openid</code> scope value to indicate that an application intends to use the OIDC protocol.</p> <p>This will return the sub claim, which uniquely identifies your user.</p>
email	Optional	<p>Returns the <code>email</code> claim, which contains:</p> <ul style="list-style-type: none"> <li>your user's email address</li> <li><code>email_verified</code>, which is a boolean indicating whether your user has verified their email address or not</li> </ul>
phone	Optional	<p>Returns the <code>phone_number</code> claim, which contains:</p> <ul style="list-style-type: none"> <li>your user's phone number</li> <li><code>phone_number_verified</code>, which is a boolean indicating whether your user has verified their phone number or not</li> </ul>
wallet-subject-id	Optional (required to use GOV.UK Wallet)	<p>Returns the <code>walletSubjectId</code> claim, which is a pairwise identifier that GOV.UK Wallet uses when it issues a credential. By comparing the returned value with the value GOV.UK Wallet submits when requesting a credential, you can be sure that the user logged into your service and GOV.UK Wallet are the same user.</p> <p>You must include this scope if you plan to <a href="#">onboard with GOV.UK Wallet (<a href="https://docs.wallet.service.gov.uk/before-integrating.html#onboard-with-gov-uk-one-login">https://docs.wallet.service.gov.uk/before-integrating.html#onboard-with-gov-uk-one-login</a>)</a> after you have onboarded with GOV.UK One Login.</p> <p>The value is returned in the format:  <code>urn:fdc:wallet.account.gov.uk:2024:3c_jJtXcLttICSNrkW7M1v02_w-SMDm2nrHsZpWQQ9</code></p> <p>where the part after <code>urn:fdc:</code> is <a href="#">Base 64 Encoding with URL and Filename Safe Alphabet</a> (<a href="https://datatracker.ietf.org/doc/html/rfc4648#section-5">https://datatracker.ietf.org/doc/html/rfc4648#section-5</a>) of the output from a SHA256 hash function.</p>

## Choose which claims your service can request

You can also request specific claims from GOV.UK One Login, if you need more information than the scopes in the previous section can provide. You must [choose a level of identity confidence \(/before-integrating/choose-the-level-of-identity-confidence/\)](#) P2 or above, otherwise you will not receive any claims in the authorisation response.

You can find details of the claims in the following table.

Claim	Description
<a href="https://vocab.account.gov.uk/v1/coreIdentityJWT">https://vocab.account.gov.uk/v1/coreIdentityJWT</a>	This claim contains core identity information about your user: <ul style="list-style-type: none"><li>their names</li><li>their date of birth</li><li>the level of confidence GOV.UK One Login reached in your user's identity</li></ul>
<a href="https://vocab.account.gov.uk/v1/address">https://vocab.account.gov.uk/v1/address</a>	This claim contains your user's postal addresses.
<a href="https://vocab.account.gov.uk/v1/passport">https://vocab.account.gov.uk/v1/passport</a>	This claim contains your user's passport details if GOV.UK One Login proved their identity using their passport.  If GOV.UK One Login did not prove your user's identity using their passport, the authorisation response will not return this claim.
<a href="https://vocab.account.gov.uk/v1/drivingPermit">https://vocab.account.gov.uk/v1/drivingPermit</a>	This claim contains your user's driving licence details if GOV.UK One Login proved their identity using their driving licence.  If GOV.UK One Login did not prove your user's identity using their driving licence, the authorisation response will not return this claim.
<a href="https://vocab.account.gov.uk/v1/returnCode">https://vocab.account.gov.uk/v1/returnCode</a>	This claim gives information about any issues with the evidence your user provided to prove their identity, for example, if GOV.UK One Login was not able to prove your user's identity. This will display as a letter code, for example <code>[{"code": "C"}]</code> , in the response.  For security reasons, you'll have to contact GOV.UK One Login on <a href="mailto:govuk-one-login@digital.cabinet-office.gov.uk">govuk-one-login@digital.cabinet-office.gov.uk</a> for more detailed information on what issue each return code represents.  If you do not include this claim in your request, GOV.UK One Login returns an <code>access_denied</code> error instead. There's further <a href="#">guidance on the <code>resultCode</code> claim (/integrate-with-integration-environment/prove-users-identity/#understand-your-user-s-return-code-claim)</a> .

You can see more about the structure of this information when you [prove your user's identity \(/integrate-with-integration-environment/prove-users-identity/\)](#).

You can only ask us for claims that are covered by your [Data Protection Impact Assessment](#) (<https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/data-protection-impact-assessments-dpias/>). You must clearly explain to your users why you are collecting the data and what you will use it for.

Once you have chosen which attributes your service can request, you can [create a configuration for each service you're integrating](#) (</before-integrating/create-individual-configurations-for-each-service/>).

This page was last reviewed on 12 February 2025.

[View source](#) [Report problem](#) [GitHub Repo](#)



[Accessibility](#)

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Choose your sector identifier

The sector identifier is a uniform resource identifier (URI) which GOV.UK One Login uses to create a pairwise user identifier, called the ‘subject identifier’.

This means you can use the sector identifier to:

- share users across multiple services
- explicitly prevent services from sharing users

You must set the sector identifier when you [register your service with GOV.UK One Login \(/before-integrating/register-and-manage-your-service/\)](#).

 **Do not change the sector identifier once your service has started to sign up or migrate users. It will change the subject identifiers GOV.UK One Login creates for each individual user.**

If you’re not sure whether you want to share users across services when onboarding your first service, you should use a generic sector identifier. Once your second service onboards, you must decide whether your services will share users.

## Set your sector identifier

Make sure your sector identifier:

- accurately represents your service or services which share users
- does not contain path information

For example, you should use `https://do-a-thing.service.gov.uk` not `https://service.gov.uk/do-a-thing`. GOV.UK One Login only uses the host part of the URI.

The following table shows an example of how to set your sector identifier using the (fictional) Department of Mythical Creatures, which has 3 services:

- tax your dragon

- register your hydra
- report a unicorn

User sharing	How to set your sector identifier	Example
Share users across all services	Set the sector identifier in all services to the same value.	Set <a href="https://mythical-creatures.gov.uk">https://mythical-creatures.gov.uk</a> as the <code>sector_identifier_uri</code> for all 3 services.
Prevent services from sharing users	Set the sector identifier in each service to a different value.	Set a separate <code>sector_identifier_uri</code> for each service: <ul style="list-style-type: none"> <li>• <a href="https://register-your-hydra.mythical-creatures.gov.uk">https://register-your-hydra.mythical-creatures.gov.uk</a></li> <li>• <a href="https://tax-your-dragon.mythical-creatures.gov.uk">https://tax-your-dragon.mythical-creatures.gov.uk</a></li> <li>• <a href="https://report-a-unicorn.mythical-creatures.gov.uk">https://report-a-unicorn.mythical-creatures.gov.uk</a></li> </ul>
Share users across some services	<p>Set the sector identifier in the services that share users to the same value.</p> <p>Give the services that should not share users a different sector identifier.</p>	<p>Set <a href="https://mythical-creatures.gov.uk">https://mythical-creatures.gov.uk</a> as the <code>sector_identifier_uri</code> for ‘register your hydra’ and ‘report a unicorn’ to share users.</p> <p>Set <a href="https://tax-your-dragon.mythical-creatures.gov.uk">https://tax-your-dragon.mythical-creatures.gov.uk</a> as the <code>sector_identifier_uri</code> for ‘tax your dragon’ to have a separate user base.</p>

This page was last reviewed on 9 February 2024.



## Accessibility

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Create a configuration for each service you're integrating

GOV.UK One Login is an OpenID Connect (OIDC) provider. An OIDC ‘relying party’ is a client application that outsources its user authentication function to an identity provider, which in this instance is GOV.UK One Login.

To interact with GOV.UK One Login, you must first [register each of your services with GOV.UK One Login \(/before-integrating/register-and-manage-your-service/\)](#). You need to do this for each of the services that you want to integrate with GOV.UK One Login.

## Understanding the client identifier

The [client identifier \(https://datatracker.ietf.org/doc/html/rfc6749#section-2.2\)](https://datatracker.ietf.org/doc/html/rfc6749#section-2.2) is a unique value GOV.UK One Login requires to identify your services. GOV.UK One Login generates the client identifier for each of your services, when you [register your service with GOV.UK One Login \(/before-integrating/register-and-manage-your-service/\)](#). GOV.UK One Login uses the client identifier to:

- retrieve configurations
- audit events
- capture performance analytics
- perform fraud prevention and data sharing

## Why you should use a specific configuration for each service

You must use individual configurations for each of your services to get the following benefits:

- service specific reports with information about success rates and volumes
- protection for each service if another service has an outage - your other services will not be affected

- effective monitoring and detection of fraudulent activity
- better help for your users because the support team will have more detailed information on user activity

If you do not use individual configurations for each of your services, GOV.UK One Login cannot:

- monitor or detect fraudulent activities as effectively
- give you service specific analytics - we cannot generate this retrospectively if you later switch to individual configurations
- provide your users with a simpler and more personalised user journey

Organisations with multiple services may have additional requirements such as:

- sharing users across services - to enable this, [set up a common sector identifier \(/before-integrating/choose-your-sector-identifier/\)](#)
- users that want to switch between services - to support users switching between services, your service must call the [/authorize](#) endpoint each time a user requests access to a new service

Once you have chosen which attributes your service can request, you can [set up your service's configuration with GOV.UK One Login \(/before-integrating/register-and-manage-your-service/\)](#).

This page was last reviewed on 15 September 2023.

[View source](#) [Report problem](#) [GitHub Repo](#)



[Accessibility](#)

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Integrating third-party platforms with GOV.UK One Login

If you use a third-party platform (Software as a Service such as Salesforce or Microsoft Power Platform, or an identity provider such as Amazon Cognito or ForgeRock) to integrate with GOV.UK One Login, you might experience issues or specific limitations during integration.

Contact GOV.UK One Login at [govuk-one-login@digital.cabinet-office.gov.uk](mailto:govuk-one-login@digital.cabinet-office.gov.uk) ([govuk-one-login@digital.cabinet-office.gov.uk](mailto:govuk-one-login@digital.cabinet-office.gov.uk) will open a link to your mail client) if you're using a third-party platform to integrate with GOV.UK One Login.

GOV.UK One Login will update this page with information on integrating with third-party platforms.

Platform	How to integrate with GOV.UK One Login
Amazon Cognito	<p>There's <a href="#">guidance on configuring Amazon Cognito to use GOV.UK One Login as an external OpenID Connect provider</a> (<a href="https://github.com/govuk-one-login/onboarding-examples/tree/main/clients/amazon-cognito">https://github.com/govuk-one-login/onboarding-examples/tree/main/clients/amazon-cognito</a>) (opens separate repository).</p>
Salesforce	<p>You'll need to build an authentication provider plugin to integrate using Salesforce.</p> <p>There's further <a href="#">guidance on building an authentication provider plugin</a> (<a href="https://github.com/govuk-one-login/onboarding-examples/blob/main/clients/salesforce-apex/README.md">https://github.com/govuk-one-login/onboarding-examples/blob/main/clients/salesforce-apex/README.md</a>) (opens separate repository).</p>

## Set up client secret using `client_secret_post`

You should only use `client_secret_post` as the token authentication method if:

- you only require authentication – `client_secret_post` is not supported for identity proving
- your third-party platform cannot support `private_key_jwt`

Contact GOV.UK One Login at [govuk-one-login@digital.cabinet-office.gov.uk](mailto:govuk-one-login@digital.cabinet-office.gov.uk) ([govuk-one-login@digital.cabinet-office.gov.uk](mailto:govuk-one-login@digital.cabinet-office.gov.uk) will open a link to your mail client) if you need to use `client_secret_post`.

You'll use OpenSSL to generate a client secret and share the hashed version of the secret with the GOV.UK One Login onboarding team.

If using `client_secret_post`, whenever you make a request to the `/token` endpoint you'll need to use the existing parameters and also add the following parameters to the token request:

- `client_id`
- `client_secret`

## Install OpenSSL

To install OpenSSL, the command will change depending on your operating system.

For macOS:

1. Follow the documentation to install [Homebrew](https://brew.sh/) (<https://brew.sh/>).
2. Run `brew install openssl`.

For Windows:

1. Follow the documentation to install [Chocolatey](https://chocolatey.org/install) (<https://chocolatey.org/install>).
2. Run `choco install openssl`.

To test if your installation has been successful, run `openssl version`.

## Generate the client secret and the salt using OpenSSL

1. Generate the client secret by running `openssl rand 40 | openssl base64 -A -out CLIENT_SECRET.txt`.
2. Generate the salt by running `openssl rand 64 | openssl base64 -A -out SALT.txt`.
3. Store the plaintext client secret (`CLIENT_SECRET.txt`) in your preferred vault following your internal standards for handling sensitive data and following the [NCSC cloud security guidance on protecting secrets](https://www.ncsc.gov.uk/collection/cloud-security-guidance-on-protecting-secrets) ([https://www.ncsc.gov.uk/collection/cloud-using-cloud-services-securely/using-a-cloud-platform-securely#section\\_11](https://www.ncsc.gov.uk/collection/cloud-using-cloud-services-securely/using-a-cloud-platform-securely#section_11)).
4. Store the plaintext salt (`SALT.txt`) on your local machine as you'll need this later.

You'll configure this plaintext secret into your application so it is available at runtime.

## Hash your client secret

You need to hash your client secret. What tooling you use to do this is up to you.

Check the following parameters are in place:

- iterations: 2
- memory: 15360
- parallelism: 1
- hash length: 16
- type: Argon2id
- output format: encoded hash

## Email the Argon2id formatted string to GOV.UK One Login

1. Open a new email and leave the email subject blank.
2. Send an email to [govuk-one-login@digital.cabinet-office.gov.uk](mailto:govuk-one-login@digital.cabinet-office.gov.uk) ([govuk-one-login@digital.cabinet-office.gov.uk](mailto:govuk-one-login@digital.cabinet-office.gov.uk) will open a link to your mail client), pasting the Argon2id encoded hash into the email body.

It's important there is no identifying information a malicious attacker could use. Make sure the email body contains only the hashed secret. Do not include:

- an email subject
- any attachments
- your client ID
- any reference for what this string is or what it is used for

If your email includes any additional information apart from the hashed secret, GOV.UK One Login will not use the secret and you'll have to create a new one.

This page was last reviewed on 4 July 2024.



## Accessibility

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Register and manage your service

You get a unique client ID when you register your service. You'll need this client ID to integrate each of your services with GOV.UK One Login.

You should configure a client ID for each environment you have. For example, if you have staging, user acceptance testing, integration and production you should configure 4 client IDs. There's further [guidance on creating a configuration for each service you're integrating \(/before-integrating/create-individual-configurations-for-each-service/#understanding-the-client-identifier\)](#).

Registering should take 5 minutes to complete. To register your service to use GOV.UK One Login, you'll need:

- a government email address
- a mobile phone

If you do not have a government email address or mobile phone, you should find a civil servant in your team who can register the service on your behalf.

Whoever registers the service will have the entry tied to their email address. It is currently not possible to reassign ownership if someone leaves or to add multiple email addresses to a particular client. If you need access after someone has left, you can create an additional client using a different email address and transfer the configuration settings to the new account.

1. Go to the [Get started with GOV.UK One Login](#) (<https://www.sign-in.service.gov.uk/getting-started>) page and select *Create admin tool account*.
2. Then, follow the on-screen instructions to enter your email address and confirm your email security code.
3. Enter your mobile number and confirm your mobile security code.
4. Fill in your client configuration details using this table.

Name	Description
Client ID	GOV.UK One Login will assign your service a unique Client ID which you must configure into your service.

Client name (Service name)	Choose the name of your service. This will be visible to your users in the sign in journey.  Choose your client name.  The client name will appear in the user interface when GOV.UK One Login redirects your user back to your service so choose something your users would recognise.  There's further <a href="https://www.gov.uk/service-manual/design/naming-your-service">guidance on naming your service</a> ( <a href="https://www.gov.uk/service-manual/design/naming-your-service">https://www.gov.uk/service-manual/design/naming-your-service</a> ).
Contacts	Enter the email addresses of your service's technical contacts – this can be a group email or multiple separate email addresses, or a combination of both.
Redirect URLs	The URL we will return your user to after they complete their GOV.UK One Login journey.  You can enter more than one URL.
Post-logout URLs	If you want to redirect your users after they log out, input one or more URLs. These will be where you redirect your users to after you have logged them out.  There's further guidance on <a href="#">logging your user out of GOV.UK One Login</a> ( <a href="#">/integrate-with-integration-environment/managing-your-users-sessions/#log-your-user-out-of-gov-uk-one-login</a> ).
Back channel logout URI	If you want to receive logout notifications from GOV.UK One Login, specify the URI of the endpoint you want GOV.UK One Login to call.  There's further guidance on <a href="#">requesting logout notifications from GOV.UK One Login</a> ( <a href="#">/integrate-with-integration-environment/managing-your-users-sessions/#request-logout-notifications-from-gov-uk-one-login</a> ).
Landing Page URL	It's not possible to configure this yet.  Send an email to <a href="mailto:govuk-one-login@digital.cabinet-office.gov.uk">govuk-one-login@digital.cabinet-office.gov.uk</a> if you need to configure this.
Sector identifier URI	Specify your service's sector identifier.  You must not change the sector identifier once your service has started to sign up or migrate users. Doing this will change the subject identifiers GOV.UK One Login creates for each individual user.

There's further [guidance on choosing your sector identifier \(/before-integrating/choose-your-sector-identifier/\)](#).

If your service has more than one `redirect_uri`, you must set the sector identifier in line with the [OpenID Connect Core 1.0 specification](#) ([https://openid.net/specs/openid-connect-core-1\\_0.html#PairwiseAlg](https://openid.net/specs/openid-connect-core-1_0.html#PairwiseAlg)).

---

**Scopes** Enter the scopes your service requires. You must include the `openid` scope.

You may choose one or more of the following:

- `email`
- `phone`

There's further [guidance on choosing which user attributes your service can request \(/before-integrating/choose-which-user-attributes-your-service-can-request/#choose-which-scopes-your-service-can-request\)](#).

---

**Claims** If you're requesting identity verification, you must include `https://vocab.account.gov.uk/v1/coreIdentityJWT`. We recommend also including `https://vocab.account.gov.uk/v1/returnCode` to make your error handling more clear. There's further [guidance on return codes](#) (<https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/prove-users-identity/#understand-your-user-s-return-code-claim>). In addition, you can choose one or more of the following:

- `https://vocab.account.gov.uk/v1/passport`
- `https://vocab.account.gov.uk/v1/drivingPermit`
- `https://vocab.account.gov.uk/v1/address`

There's further guidance on [choosing which claims your service can request \(/before-integrating/choose-which-user-attributes-your-service-can-request/#choose-which-claims-your-service-can-request\)](#).

---

Token Authentication method	Specify the token authentication method your service is using. This will be <code>private_key_jwt</code> or <code>client_secret_post</code> .  There's further <a href="#">guidance on using the correct token authentication method for your service (/before-integrating/use-correct-token-authentication-method/)</a> .
Public key	Only include this if your service is using the <code>private_key_jwt</code> token authentication method.  Enter the contents of your public key Privacy Enhanced Mail (PEM) file (or whichever file was created when you created your key pair).  There's further <a href="#">guidance on generating a key pair (/before-integrating/set-up-your-public-and-private-keys/#create-a-key-pair)</a> .
ID token signing algorithm	Choose either <code>RS256</code> or <code>ES256</code> .  By default, GOV.UK One Login will sign the <code>id_token</code> JSON Web Token (JWT) using the <code>ES256</code> algorithm but some third party tooling does not support <code>ES256</code> . If your service needs an alternative algorithm, we can sign your <code>id_token</code> JWT using the <code>RS256</code> algorithm

This page was last reviewed on 25 September 2024.

[View source](#) [Report problem](#) [GitHub Repo](#)



## Accessibility

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Set up your public and private keys

GOV.UK One Login uses public key cryptography to authenticate services, so you'll need to create a key pair (a public key and a corresponding private key). Then you'll need to [share your public key with GOV.UK One Login \(/before-integrating/set-up-your-public-and-private-keys/#share-your-public-key-with-gov-uk-one-login\)](#) when registering your service.

You'll also need to use your private key when:

- you're registering your service to use GOV.UK One Login environments, such as integration or production
- you request the token using the private key authentication mechanism on the [/token](#) endpoint

## Create a key pair

You can create a key pair using [OpenSSL \(https://www.openssl.org/\)](#). After you've installed OpenSSL, run the following on your command line to create your key pair:

```
openssl genpkey -algorithm RSA -out private_key.pem -pkeyopt rsa_keygen_bits  
openssl rsa -pubout -in private_key.pem -out public_key.pem
```

You have now created your key pair, which will appear on your machine as 2 files:

- [public\\_key.pem](#) - this is your public key, which you should share with GOV.UK One Login
- [private\\_key.pem](#) - this is your private key, which you should store securely and not share

 Once you've generated your private key, you must store the key in a secure location, such as a file vault. You must not share your private key.

# Share your public key with GOV.UK One Login

Once you've created your key pair, share your public key with GOV.UK One Login. You have 2 options to do this:

- [share a fixed public key \(/before-integrating/set-up-your-public-and-private-keys/#share-a-fixed-public-key\)](#) directly - if you use a fixed public key and start signing with a new key before GOV.UK One Login updates your service's configuration, users will not be able to access your service with GOV.UK One Login
- (recommended) [share your public key\(s\) using a JSON Web Key Set \(JWKS\) endpoint \(/before-integrating/set-up-your-public-and-private-keys/#share-your-public-keys-using-a-jwks-endpoint\)](#)

We recommend using a JWKS endpoint to share your public keys. A JWKS endpoint is a read-only URL that returns JWKSs as JSON objects so you can share multiple public keys. If you do this, you can rotate your keys without contacting GOV.UK One Login for a configuration change. You can update the JWKS endpoint to contain both the old and new keys, then immediately start signing with the new key. This means users can still access your service with GOV.UK One Login.

## Share a fixed public key

If you're using a fixed public key, send the public key you created to GOV.UK One Login. You can check what to send when you [contact the GOV.UK One Login team to register your service \(/before-integrating/register-and-manage-your-service/\)](#).

## Share your public keys using a JWKS endpoint

If you're using a JWKS endpoint, you'll need to make sure it works with GOV.UK One Login.

This means your endpoint must:

- use HTTPS
- be publicly accessible
- return a **HTTP 200 (OK)** within 5 seconds of a GET request
- return an RSA signing key in JWKS format
- return a unique `kid` parameter in each key (`JWK`) entry
- include the `kid` parameter for the key used to sign a `JWS` in its header

Your JWKS endpoint should give a JSON response similar to the following example:

```
{  
  "keys": [  
    {  
      "kty": "RSA",  
      "e": "AQAB",  
      "use": "sig",  
      "kid": "f58a6bef-0d22-444b-b4d3-507a54e9892f",  
      "n": "pSx43eUV2hZ3AJKYNFHx0sILQ_tUNpfPVELCy3js3FsTp5Mcbpb8mu-arekTCq0M  
    }  
  ]  
}
```

Once you have shared a JWKS endpoint URL, you can [choose which user attributes your service can request](#) ([/before-integrating/choose-which-user-attributes-your-service-can-request/](#)).

## Revoking a public key on your JWKS endpoint

Contact GOV.UK One Login if you need to immediately revoke a public key on your JWKS endpoint.

GOV.UK One Login caches keys for up to 24 hours, so do not remove a compromised key from your JWKS endpoint without also telling GOV.UK One Login.

This page was last reviewed on 22 August 2024.

[View source](#) [Report problem](#) [GitHub Repo](#)



## Accessibility

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Setting a User-Agent header on HTTP requests

When your service calls GOV.UK One Login directly it is important to ensure your HTTP client sets a [User-Agent](https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Headers/User-Agent) (<https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Headers/User-Agent>) header in all HTTP requests.

Failure to set the required header will result in your service receiving a [403 Forbidden](#) error message from the GOV.UK One Login service.

Most libraries and platforms do this automatically, but some do not. Your User-Agent header should contain your platform or library and the version, for example [curl/7.64.1](#) or [PostmanRuntime/7.26.5](#).

You can further augment this by identifying your service, by including the service URL, as this could be useful when debugging any issues you be encountering, for example:

```
User-Agent: my-platform/version (https://my-service-url.gov.uk)
```

The value you choose should be consistently used on every direct call to the GOV.UK One Login.

This page was last reviewed on 22 October 2025.

[View source](#) [Report problem](#) [GitHub Repo](#)



## Accessibility

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Use the correct token authentication method

The platform you use to integrate with GOV.UK One Login will affect which token authentication method you need to use.

Most services will use `private_key_jwt`. However, if you're using a third-party platform which does not support `private_key_jwt`, you may be granted an exception to use `client_secret_post`.

You can [read more guidance on third-party platforms](#) ([/before-integrating/integrating-third-party-platform/](#)) to learn about which ones do not support `private_key_jwt`.

This page was last reviewed on 4 July 2024.

[View source](#) [Report problem](#) [GitHub Repo](#)



[Accessibility](#)

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Before you integrate with GOV.UK One Login

When you plan your integration with GOV.UK One Login, you should consider:

- how many services within your organisation you're planning to integrate
- if your services need to share users, in case you're integrating more than 1 service
- if you need to create a reusable component to standardise integration across your organisation, in case you're integrating a large number of services
- what the scope of your individual services is and whether this meets the GOV.UK Service Standard definition of a service

Make sure you scope your services according to the [GOV.UK Service Standard guidance](#) (<https://www.gov.uk/service-manual/service-standard>) on how users think and what they need to do. Find more [information on scoping your service](#) (<https://www.gov.uk/service-manual/design/scoping-your-service>).

Before you can start integrating with GOV.UK One Login, you need to:

- [choose the level of authentication for your service](#) (</before-integrating/choose-the-level-of-authentication/#choose-the-level-of-authentication-for-your-service>)
- [choose the level of identity confidence for your service](#) (</before-integrating/choose-the-level-of-identity-confidence/>)
- [generate a key pair](#) (</before-integrating/set-up-your-public-and-private-keys/#create-a-key-pair>)
- [choose which user attributes your service can request](#) (</before-integrating/choose-which-user-attributes-your-service-can-request/>)
- [create a configuration for each service you're integrating](#) (</before-integrating/create-individual-configurations-for-each-service/>)
- [set up your service's configuration with GOV.UK One Login](#) (</before-integrating/register-and-manage-your-service/>)
- [ensure you are setting the User-Agent header on calls to GOV.UK One Login](#) (</before-integrating/set-user-agent-header/>)

To get started, you'll need to [choose the level of authentication for your service](#) (</before-integrating/choose-the-level-of-authentication/#choose-the-level-of-authentication-for-your-service>).

This page was last reviewed on 15 September 2023.

[View source](#) [Report problem](#) [GitHub Repo](#)



[Accessibility](#)

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Configure your service for production

 **You must configure your service for production at least 2 weeks before you start using the production environment in private beta or public beta.**

Before you can configure your service for production, you must [integrate with GOV.UK One Login's integration environment \(/integrate-with-integration-environment/\)](#).

1. Tell your Engagement Manager that you need to configure your service in production – if you do not have an Engagement Manager, [complete the form to register your interest \(<https://www.sign-in.service.gov.uk/register>\)](#). You'll need to complete this form a minimum of 6 weeks before your go-live date.
2. You only need to send your Engagement Manager the service name and client ID of the client you've been testing in your integration configuration. The GOV.UK One Login team will send you a draft configuration in JSON format including the new client ID for your production service.
3. Update the JSON configuration by replacing the placeholder values with your service's configuration. There's [guidance on understanding the JSON configuration \(/configure-for-production/#use-the-table-to-understand-the-json-configuration\)](#).
4. Send your modified JSON configuration back to your Engagement Manager by email. The GOV.UK One Login team will check your production configuration and contact you if we need more information.
5. Configure the new client ID into your own application code and deploy to your production environment.
6. Test your application works in production. This could be a limited test with a small number of users or a limited private beta.

## Use the table to understand the JSON configuration

Field	Notes
Field	Notes

BackChannelLogoutUri	If you want to receive logout notifications from GOV.UK One Login, specify the production URI of the endpoint you want GOV.UK One Login to call.  This must be a production-grade URI with domains without reference to <code>http://</code> and <code>localhost</code> .  There's further guidance on <a href="#">requesting logout notifications from GOV.UK One Login (/integrate-with-integration-environment/managing-your-users-sessions/#request-logout-notifications-from-gov-uk-one-login)</a> .
ClientID	GOV.UK One Login will fill in <code>ClientID</code> with your production client ID. You do not need to do anything.
Claims	If you're doing identity verification, you'll need to specify which claims your service requires. You may choose one or more of the following: <ul style="list-style-type: none"><li><code>https://vocab.account.gov.uk/v1/passport</code></li><li><code>https://vocab.account.gov.uk/v1/drivingPermit</code></li><li><code>https://vocab.account.gov.uk/v1/coreIdentityJWT</code></li><li><code>https://vocab.account.gov.uk/v1/address</code></li><li><code>https://vocab.account.gov.uk/v1/returnCode</code></li></ul>
ClientName	Choose your client name. The client name will appear in the user interface when GOV.UK One Login redirects your user back to your service so choose something your users would recognise.  There's further <a href="#">guidance on naming your service (https://www.gov.uk/service-manual/design/naming-your-service)</a> .
ClientType	Leave this field as <code>web</code> .
ConsentRequired	Leave this field as <code>false</code> .
ContactS	Enter your service's technical contact email addresses – this can be a group email or multiple separate email addresses, or a combination of both.
CookieConsentShared	Leave this field as <code>false</code> .

IdentityVerificationSupported	If you're using identity verification, this should be <code>true</code> . If you only need authentication, this should be <code>false</code> .
IdTokenSigningAlgorithm	This will be <code>ES256</code> or <code>RS256</code> . You can find the one you're using in your application's code.
LandingPageUrl	<p><code>LandingPageUrl</code> is only required if you're making identity requests.</p> <p>GOV.UK One Login supports a single <code>LandingPageUrl</code> after a user returns from an offline journey. Specify the production URL your user will be redirected to after they visit the Post Office. This link will allow them to continue their sign up process for your service.</p> <p>These must be production-grade URLs without reference to <code>http://</code> and <code>localhost</code>.</p>
OneLoginService	Leave this field as <code>false</code> .
PostLogoutRedirectUris	<p>If you want to redirect your users after they log out, input one or more production URLs. These will be where you redirect your users to after you have logged them out.</p> <p>These must be production-grade URLs without reference to <code>http://</code> and <code>localhost</code>.</p> <p>There's further guidance on <a href="#">logging your user out of GOV.UK One Login (/integrate-with-integration-environment/managing-your-users-sessions/#log-your-user-out-of-gov-uk-one-login)</a>.</p>
PublicKey	<p><code>PublicKey</code> is only required if you're using the <code>private_key_jwt</code> token authentication method.</p> <p>Enter the contents of your public key Privacy Enhanced Mail (PEM) file (or whichever file was created when you created your key pair).</p> <p>There's further <a href="#">guidance on generating a key pair (/before-integrating/set-up-your-public-and-private-keys/#create-a-key-pair)</a>.</p>
IsInternalServer	Leave this field as <code>false</code> .

---

JarValidationRequired	GOV.UK One Login will fill in this field.
RedirectUrls	Enter one or more of your service's production redirect URLs. These must be production-grade URLs without reference to <code>http://</code> and <code>localhost</code> .
Scopes	Enter the scopes your service requires. You must include the <code>openid</code> scope.  You may choose one or more of the following: <ul style="list-style-type: none"><li>• <code>email</code></li><li>• <code>phone</code></li></ul> There's further <a href="#">guidance on choosing which user attributes your service can request</a> ( <a href="#">/before-integrating/choose-which-user-attributes-your-service-can-request/#choose-which-scopes-your-service-can-request</a> ).
SectorIdentifierUri	Specify your service's sector identifier.  You must not change the sector identifier once your service has started to sign up or migrate users. Doing this will change the subject identifiers GOV.UK One Login creates for each individual user.  There's further <a href="#">guidance on choosing your sector identifier</a> ( <a href="#">/before-integrating/choose-your-sector-identifier/</a> ).  If your service has more than one <code>redirect_uri</code> , you must set the sector identifier in line with the <a href="#">OpenID Connect Core 1.0 specification</a> ( <a href="https://openid.net/specs/openid-connect-core-1_0.html#PairwiseAlg">https://openid.net/specs/openid-connect-core-1_0.html#PairwiseAlg</a> ).
ServiceType	Leave this field as <code>MANDATORY</code> .
SubjectType	Leave this field as <code>pairwise</code> .
TestClient	Leave this field as <code>false</code> .
TokenAuthenticationMethod	Specify the token authentication method your service is using. This will be <code>private_key_jwt</code> or <code>client_secret_post</code> .

There's further [guidance on using the correct token authentication method for your service](#) (/before-integrating/use-correct-token-authentication-method/).

---

**PKCEEnforced** Specify whether your service must provide the parameters required for Proof Key for Code Exchange (PKCE) protocol in the [authorize](#) (/integrate-with-integration-environment/authenticate-your-user/#make-a-request-to-the-authorize-endpoint) and [token](#) (/integrate-with-integration-environment/authenticate-your-user/#make-a-token-request) requests.

You can [read more about PKCE in RFC 7636](#) (<https://datatracker.ietf.org/doc/html/rfc7636>).

---

## Use the production discovery endpoint

You can use the [production discovery endpoint](#) (<https://oidc.account.gov.uk/.well-known/openid-configuration>) (viewed at <https://oidc.account.gov.uk/.well-known/openid-configuration>).

This page was last reviewed on 20 February 2025.

[View source](#) [Report problem](#) [GitHub Repo](#)

---



[Accessibility](#)

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Error messages

This page collates the error messages from GOV.UK One Login.

## Error messages from the /authorize endpoint

Error	More information about your error
<code>unauthorized_client</code>	In rare circumstances, such as a security incident, One Login may prevent users from logging in to your service. If this happens, the error code <code>unauthorized_client</code> will be returned with the error description <code>client deactivated</code> . When your service receives this error, you must show the user a custom error page to explain that they cannot use your service at the moment and should try again later.
<code>request_is_missing_parameters</code>	<p>The request has one or more of the following issues:</p> <ul style="list-style-type: none"><li>missing a required parameter</li><li>includes an invalid parameter value</li><li>includes a parameter more than once</li><li>not in the correct format</li></ul> <p>You should <a href="#">check you have included the correct parameters (/integrate-with-integration-environment/authenticate-your-user/#replace-the-placeholder-values-in-your-example)</a>, especially the <code>client_id</code>, <code>redirect_uri</code>, <code>response_type</code> and <code>scope</code> parameters.</p>

<code>invalid_request</code>	<p>The request has one or more of the following issues:</p> <ul style="list-style-type: none"><li>• missing a required parameter</li><li>• includes an invalid parameter value</li><li>• includes a parameter more than once</li><li>• not in the correct format</li></ul> <p>. You should <a href="#">check you have included the correct parameters (/integrate-with-integration-environment/authenticate-your-user/#replace-the-placeholder-values-in-your-example)</a>, especially the <code>client_id</code>, <code>redirect_uri</code>, <code>response_type</code> and <code>scope</code> parameters.</p> <hr/>
<code>invalid_request - Request vtr not valid</code>	<p>You've requested single factor authentication and identity information. To make a successful identity request, you must request two-factor authentication and the identity level of confidence, for example <a href="#">C1.Cm.P2</a>.</p>
<code>invalid_scope</code>	<p>The scope or scopes you have requested are invalid, unknown, or are not in the correct format.</p> <p>You can read more about scopes in <a href="#">choosing which user attributes your service can request (/before-integrating/choose-which-user-attributes-your-service-can-request/)</a>.</p> <hr/>
<code>unsupported_response_type</code>	<p>Your service is not registered for the requested <code>response_type</code>. You must set the <code>response_type</code> to be code: <code>response_type=code</code>.</p>
<code>server_error</code>	<p>The GOV.UK One Login authentication server has experienced an internal server error.</p> <hr/>
<code>temporarily_unavailable</code>	<p>If you're only making an authentication request (as opposed to requesting both authentication and identity), this error code means the GOV.UK One Login authentication server is temporarily unavailable, which might be caused by temporary overloading or planned maintenance. Make your request again in a few minutes.</p> <p>If you're making an identity request and you get this error, it means the identity proving and verification does not currently have capacity for this request.</p> <hr/>
<code>access_denied</code>	<p>GOV.UK One Login returns this error in 2 scenarios.</p> <p>The first scenario is that the session in the user's browser is unavailable. This can happen when your user's cookies have been lost or your user changed browsers during the identity verification process. You should then</p>

[make another authentication and identity request \(<https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/authenticate-your-user/#make-a-request-for-authentication-and-identity>\)](https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/authenticate-your-user/#make-a-request-for-authentication-and-identity). You must help your user try again, for example by going back to the start of your authentication and identity verification process.

The second scenario is that the identity evidence your user provided has a lower score than the identity confidence specified in your request. As a result, GOV.UK One Login could not return the medium level of identity confidence ( P2 ) and instead returned a lower level of identity confidence.

If you're using return codes, you will not receive an error for this scenario. Find more information on [understanding the return codes claim \(\[/integrate-with-integration-environment/prove-users-identity/#understand-your-user-s-return-code-claim\]\(https://integrate-with-integration-environment/prove-users-identity/#understand-your-user-s-return-code-claim\)\)](https://integrate-with-integration-environment/prove-users-identity/#understand-your-user-s-return-code-claim).

---

<code>login_required</code>	You have made a re-authentication request and a user is unable to authenticate themselves.
-----------------------------	--

Reasons for this could be:

- the user has entered a different email address to the one stored
- the user has entered the wrong password too many times
- the user has entered the wrong one-time password (OTP) code too many times

---

## Error messages from the /userinfo endpoint

Error	More information about your error
<code>invalid_token</code>	GOV.UK One Login denied your request as you have an invalid or missing bearer access token.  To proceed, you must use the authorisation header field to send the token as a <a href="https://oauth.net/2/bearer-tokens/">bearer token (<a href="https://oauth.net/2/bearer-tokens/">https://oauth.net/2/bearer-tokens/</a>)</a> .

---

## Error messages from the /token endpoint

Error	More information about your error
invalid_request	<p>The request is missing a parameter so the server cannot proceed with the request. This error may also be returned if the request includes an unsupported parameter or repeats a parameter.</p> <p>Review your parameters and check they are supported and not repeated.</p>
invalid_client	<p>Client authentication failed, which could be caused by the request containing an invalid <code>client_id</code> or an issue in validating the signature of the <code>client_assertion</code>.</p> <p>To resolve, check:</p> <ul style="list-style-type: none"><li>• your <code>client_id</code> matches the <code>client_id</code> you received when you <a href="#">registered your service to use GOV.UK One Login (/before-integrating/register-and-manage-your-service/)</a></li><li>• you have signed your <code>client_assertion</code> JWT with the private key generated when you <a href="#">registered your service to use GOV.UK One Login (/before-integrating/register-and-manage-your-service/)</a></li><li>• your service uses a <a href="#">key signing algorithm which GOV.UK One Login supports (https://oidc.account.gov.uk/.well-known/openid-configuration)</a></li></ul>
invalid_grant	<p>The request has one or more of the following issues:</p> <ul style="list-style-type: none"><li>• the authorisation code is invalid or expired</li><li>• the redirect URL given in the authorisation request does not match the URL provided in this access token request</li><li>• the authorisation request included Proof Key for Code Exchange (PKCE) parameters, and the <code>code_verifier</code> is missing or invalid</li></ul>
unauthorized_client	<p>The application is successfully authenticated, but it's not registered to use the requested <a href="#">grant type (https://oauth.net/2/grant-types/)</a>.</p>
unsupported_grant_type	<p>The grant type is not supported by the server.</p>

This page was last reviewed on 21 January 2025.



## Accessibility

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# How GOV.UK One Login works

GOV.UK One Login is an [OpenID Connect \(OIDC\) \(https://openid.net/connect/\)](https://openid.net/connect/)-compliant service that helps you authenticate your users who are using services they've logged into with their GOV.UK One Login.

GOV.UK One Login follows the Service Manual for [designing for different browsers and devices \(https://www.gov.uk/service-manual/technology/designing-for-different-browsers-and-devices\)](https://www.gov.uk/service-manual/technology/designing-for-different-browsers-and-devices).

GOV.UK One Login uses 2 different environments:

- an integration environment, which contains sample user data (for example, date of birth, address) which you can use to test your service's integration with GOV.UK One Login
- a production environment, which is the live environment for real users to access and use your service's integration with GOV.UK One Login

## Understand the flow GOV.UK One Login uses

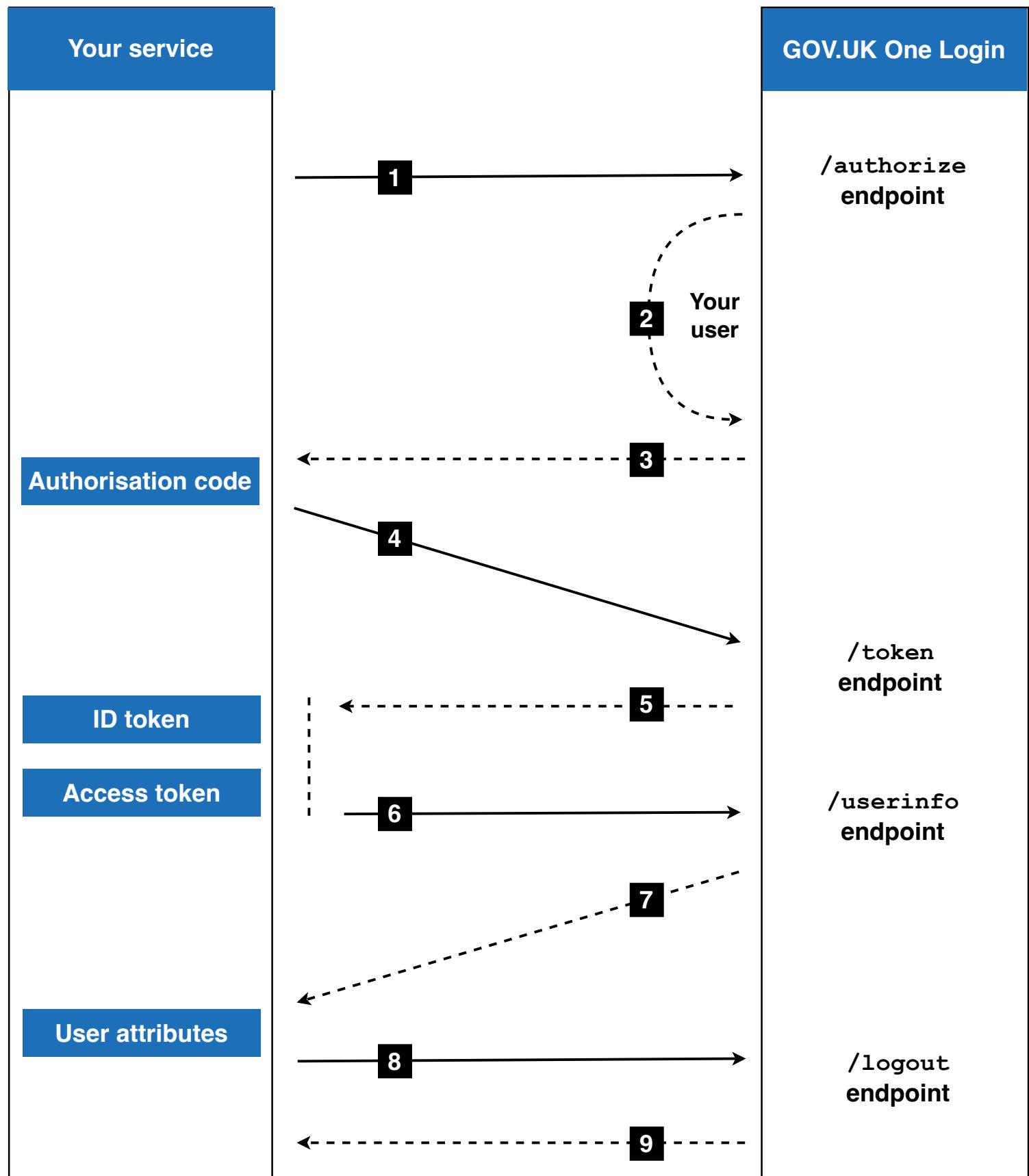


1. Your service asks the user to sign in or create an account.
2. If your service needs confidence your user is who they say they are, GOV.UK One Login will request proof of identity.
3. GOV.UK One Login collects evidence of the user's identity.
4. GOV.UK One Login provides information about your user.

You can read [guidance about cookies on GOV.UK One Login \(https://signin.account.gov.uk/cookies\)](https://signin.account.gov.uk/cookies) if you want to learn more about cookies.

To understand the technical flow, for example the endpoints, requests and tokens, there's a more detailed technical diagram you can use.

# Understand the technical flow GOV.UK One Login uses



1. Your service makes an [authorisation request](#) (`/integrate-with-integration-environment/authenticate-your-user/#make-a-request-to-the-authorize-endpoint`) to the `/authorize endpoint`.

[/authorize](#) endpoint.

2. The user logs in (or creates an account if they do not have one) and proves their identity if your service needs them to. GOV.UK One Login lets your user know how their data will be shared with your service.
3. GOV.UK One Login returns an [authorisation code \(/integrate-with-integration-environment/authenticate-your-user/#generate-an-authorisation-code\)](#) to your service.
4. Your service makes a [token request \(/integrate-with-integration-environment/authenticate-your-user/#make-a-token-request\)](#) to the [/token](#) endpoint and includes the authorisation code in the request.
5. Your service receives an [ID token and access token \(/integrate-with-integration-environment/authenticate-your-user/#receive-response-for-make-a-token-request\)](#) in the response.
6. Your service makes a request to the [/userinfo](#) endpoint to [retrieve user information \(/integrate-with-integration-environment/authenticate-your-user/#retrieve-user-information\)](#). You can read more about [choosing which user attributes your service can request \(/before-integrating/choose-which-user-attributes-your-service-can-request\)](#).
7. Your service receives a [response containing user attributes \(/integrate-with-integration-environment/authenticate-your-user/#receive-response-for-retrieve-user-information\)](#).
8. Your service makes a [log out request \(/integrate-with-integration-environment/managing-your-users-sessions/#log-your-user-out-of-gov-uk-one-login\)](#) to the [/logout](#) endpoint.
9. Your service receives an [HTTP 302](#) response redirecting the user to the [post\\_logout\\_redirect\\_uri](#).

Find out [what to consider before you integrate your service with GOV.UK One Login \(/before-integrating\)](#).

This page was last reviewed on 10 April 2024.

[View source](#) [Report problem](#) [GitHub Repo](#)



## Accessibility

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Authenticate your user

To get an access token which will allow you to access basic user information, you'll need to integrate with [OAuth's Authorization Code Flow \(https://openid.net/specs/openid-connect-core-1\\_0.html\)](https://openid.net/specs/openid-connect-core-1_0.html).

## Use the integration discovery endpoint

You can use the [integration discovery endpoint \(https://oidc.integration.account.gov.uk/.well-known/openid-configuration\)](https://oidc.integration.account.gov.uk/.well-known/openid-configuration) (viewed at <https://oidc.integration.account.gov.uk/.well-known/openid-configuration>) to get information needed to interact with GOV.UK One Login, for example:

- issuer name
- information about the keys
- supported scopes, which will contain the user attributes your service can request

When you configure your service for production, you can [use the production discovery endpoint \(/configure-for-production/#use-the-production-discovery-endpoint\)](#).

## Make a request to the /authorize endpoint

You can send a request to the [/authorize](#) endpoint to:

- authenticate your user
- check your user's level of identity confidence - you must have authenticated your user first

Choose one of the following example messages to make your own [GET](#) request. You can use the following table to [replace the placeholders in your example message \(/integrate-with-integration-environment/authenticate-your-user/#replace-the-placeholder-values-in-your-example\)](#).

### Make a request for authentication

To authenticate your user, customise the following example **GET** request by [replacing the example's placeholder values \(/integrate-with-integration-environment/authenticate-your-user/#replace-the-placeholder-values-in-your-example\)](#).

The following example specifies a medium level of authentication. There's further guidance on choosing the [level of authentication \(/before-integrating/choose-the-level-of-authentication/#choose-the-level-of-authentication-for-your-service\)](#).

```
GET /authorize?response_type=code  
&scope=YOUR_SCOPES  
&client_id=YOUR_CLIENT_ID  
&state=STATE  
&redirect_uri=YOUR_REDIRECT_URI  
&nonce=aEwkamaos5B  
&vtr=["C1.Cm"]  
&ui_locales=en  
  
HTTP/1.1  
Host: oidc.integration.account.gov.uk
```

 **This code example uses formatting that makes it easier to read. If you copy the example, you must make sure the request is:**

- **a continuous line of text separating each parameter with an ampersand (&)**
- **not split across multiple lines**
- **without any additional separators such as newline, commas or tabs**

## Make a request for authentication and identity

If you need to authenticate your user and check their identity, you should send 2 separate requests: one for authentication and one for identity.

1. [Send a request to the /authorize endpoint to authenticate your user \(/integrate-with-integration-environment/authenticate-your-user/#make-a-request-to-the-authorize-endpoint\)](#) specifying the Vector of Trust (**vtr**) parameter as **C1.Cm** .
2. Send a request for identity to the **/authorize** endpoint specifying the **vtr** as **C1.Cm.P2** .

By using 2 separate requests:

- more users are likely to create their account successfully

- you can track which users could not prove their identity
- you can support your users better when returning from an in-person identity check because you'll have authenticated them previously
- you simplify the migration of existing users to GOV.UK One Login

The following example uses medium authentication ([C1.Cm](#)) and a medium level of identity confidence ([P2](#)). There's further guidance on choosing the [level of authentication](#) ([/before-integrating/choose-the-level-of-authentication/#choose-the-level-of-authentication-for-your-service](#)) and choosing the [level of identity confidence](#) ([/before-integrating/choose-the-level-of-identity-confidence/](#)).

You can [replace your example's placeholder values](#) ([/integrate-with-integration-environment/authenticate-your-user/#replace-the-placeholder-values-in-your-example](#)).

```
GET /authorize?response_type=code
&scope=YOUR_SCOPES
&client_id=YOUR_CLIENT_ID
&state=STATE
&redirect_uri=YOUR_REDIRECT_URI
&nonce=aEwkamaos5B
&vtr=["C1.Cm.P2"]
&ui_locales=en
&claims=<claims-request>
HTTP/1.1
Host: oidc.integration.account.gov.uk
```

 **This code example uses formatting that makes it easier to read. If you copy the example, you must make sure the request is:**

- a continuous line of text separating each parameter with an ampersand (&)
- not split across multiple lines
- without any additional separators such as newline, commas or tabs

## Create a URL-encoded JSON object for <claims-request>

After you've made a request for authentication and identity, you should then create a URL-encoded JSON object for [<claims-request>](#). Your JSON object should look similar to this example:

```
{  
  "userinfo": {  
    "https://vocab.account.gov.uk/v1/coreIdentityJWT": null,  
    "https://vocab.account.gov.uk/v1/address": null,  
    "https://vocab.account.gov.uk/v1/passport": null,  
    "https://vocab.account.gov.uk/v1/drivingPermit": null,  
    "https://vocab.account.gov.uk/v1/returnCode": null  
  }  
}
```

You can only request user attributes to be returned in the `/userinfo` response. You cannot configure the claims returned in the [ID token \(/integrate-with-integration-environment/authenticate-your-user/#understand-your-id-token\)](#).

## Secure your authorisation request parameters with JWT

You can use a JWT-secured OAuth 2.0 authorisation request (JAR) with encoded parameters to protect your request from attacks and hackers.

GOV.UK One Login follows the [OIDC principles on passing request objects](#) ([https://openid.net/specs/openid-connect-core-1\\_0.html#RequestObject](https://openid.net/specs/openid-connect-core-1_0.html#RequestObject)).

1. Build a request object and sign it using the [private key you created \(/before-integrating/set-up-your-public-and-private-keys/#create-a-key-pair\)](#) when setting up your integration with GOV.UK One Login.
2. Encode the signed request object.
3. Make a `GET` request replacing `YOUR_REQUEST_OBJECT` with your signed and encoded request object.

Use this example to make your own `GET` request, replacing the placeholder values:

```
GET /authorize?response_type=code  
&scope=YOUR_SCOPES  
&client_id=YOUR_CLIENT_ID  
&request=YOUR_REQUEST_OBJECT  
HTTP/1.1  
Host: oidc.integration.account.gov.uk
```

You must make sure:

- `response_type`, `scope`, and `client_id` are identical in the query parameters and the request object
- you do not set any other OIDC parameters using query parameters

Before you encode and sign the request object, it should look similar to this example:

```
{
  "aud": "https://oidc.integration.account.gov.uk/authorize",
  "iss": "YOUR_CLIENT_ID",
  "response_type": "code",
  "client_id": "YOUR_CLIENT_ID",
  "redirect_uri": "https://client.example.org/cb",
  "scope": "YOUR_SCOPES",
  "state": "af0ifjsldkj",
  "nonce": "n-0S6_WzA2Mj",
  "vtr": [
    "Cl.Cm.P2"
  ],
  "ui_locales": "en",
  "claims": {
    "userinfo": {
      "https://vocab.account.gov.uk/v1/coreIdentityJWT": null,
      "https://vocab.account.gov.uk/v1/address": null,
      "https://vocab.account.gov.uk/v1/passport": null,
      "https://vocab.account.gov.uk/v1/drivingPermit": null
    }
  }
}
```

You can [replace your example's placeholder values \(/integrate-with-integration-environment/authenticate-your-user/#replace-the-placeholder-values-in-your-example\)](#).

## Replace the placeholder values in your example

Use the guidance in the following table to replace placeholder values in your example message.

Parameter	Required or optional	Description
-----------	----------------------------	-------------

response_type	Required	<p>You must set this value to be code: <code>response_type=code</code> . If you're using JAR, make sure the <code>response_type</code> values in the query parameters and the request object are identical.</p>
scope	Required	<p>A space-separated list of scopes. You must include <code>openid</code> as one scope value. If you request <code>openid</code> but also request other incorrect scopes, the error <code>invalid_scope</code> will return with an HTTP 302 instead.</p> <p>You should refer to the guidance on <a href="#">choosing which user attributes your service can request (/before-integrating/choose-which-user-attributes-your-service-can-request/)</a> for the <code>scope</code> parameter.</p> <p>If you're using JAR, make sure the <code>scope</code> values in the query parameters and the request object are identical.</p>
client_id	Required	<p>The <a href="#">client identifier (/before-integrating/create-individual-configurations-for-each-service/#understanding-the-client-identifier)</a>, which we generated for you when you <a href="#">registered your service to use GOV.UK One Login (/before-integrating/register-and-manage-your-service/)</a> must match your client configuration.</p> <p>If you're using JAR, make sure the <code>client_id</code> values in the query parameters and the request object are identical.</p>
state	Required	<p>When you receive a response at the redirect URL, there must be a way to verify the response came for a request which you sent. The <code>state</code> value solves this issue by binding the request and response, which reduces impact of <a href="#">Cross Site Request Forgery (https://owasp.org/www-community/attacks/csrf)</a> attacks.</p> <p>This value will be returned to the client in the authentication response.</p>
redirect_uri	Required	<p>You'll have specified your <code>redirect_uri</code> when you <a href="#">registered your service to use GOV.UK One Login (/before-integrating/register-and-manage-your-service/)</a>.</p> <p>To avoid an <code>HTTP 400 Bad Response</code> error, the redirect URI must exactly match one of the URIs configured in your client configuration and also include the protocol <code>https://</code> or <code>http</code> .</p>

If you're using request parameters, the value must be URL-encoded.

---

nonce	Required	<p>A unique value generated by your application that is used to verify the integrity of the <code>id_token</code> and mitigate replay attacks.</p> <p>This value will be present in the <code>id_token</code> and should include the per-session state, as well as being impossible for attackers to guess.</p> <p>Your application will need to verify the <code>nonce</code> claim value is the same as the <code>nonce</code> parameter sent in the authentication request.</p>
aud	Optional	<p>If you're using JAR, you must include <code>aud</code> in your JSON object.</p> <p>You must set this value to specify GOV.UK One Login's authorisation server as the intended audience:</p> <p><code>aud=https://oidc.integration.account.gov.uk/authorize</code> .</p>
iss	Optional	<p>If you're using JAR, the <code>iss</code> parameter is required.</p> <p>You must set this value to be your <code>client_id</code>. GOV.UK One Login generated your <code>client_id</code> when you <a href="#">registered your service to use GOV.UK One Login</a> (<a href="#">/before-integrating/register-and-manage-your-service/</a>).</p>
ui_locales	Optional	<p>GOV.UK One Login supports English and Welsh as language choices.</p> <p>If your service is in Welsh, you may want to display GOV.UK One Login in Welsh for a consistent user experience. You can use <code>ui_locales</code> to do this.</p> <p>In the <code>ui_locales</code> parameter, you can choose either <code>en</code> (English) or <code>cy</code> (Welsh).</p> <p>Using <code>ui_locales</code> is optional. If you do not include it, your service will continue using English by default.</p> <p>GOV.UK One Login does not support any other languages.</p>
vtr	Optional	The <code>vtr</code> parameter represents 'Vectors of Trust' where you request authentication and, optionally, identity proving. For example, if you want the medium level of authentication and medium identity confidence, request <code>vtr=[“Cl.Cm.P2”]</code> .

You selected your Vector of Trust when you [chose the level of authentication \(/before-integrating/choose-the-level-of-authentication/#choose-the-level-of-authentication-for-your-service\)](#) and [the level of identity confidence \(/before-integrating/choose-the-level-of-identity-confidence/\)](#) for your service.

You can read more about how to combine the vectors for authentication level and identity confidence in [Section 3 of RFC 8485](#) (<https://datatracker.ietf.org/doc/html/rfc8485#section-3.1>). If you need identity proving, you must request **C1.Cm** (the medium level of authentication).

If you do not specify the **vtr** parameter, your service will automatically log your users in at the medium level of authentication (**C1.Cm**). This means you will not receive identity attributes in your response.

<b>claims</b>	Optional	To get the identity attributes your service needs, you should specify these in the <b>claims</b> parameter using the <b>/userinfo</b> endpoint. The <b>/userinfo</b> endpoint returns a JSON object listing the requested claims.  You can read more about <a href="#">choosing which user attributes your service can request (/before-integrating/choose-which-user-attributes-your-service-can-request/)</a> .  You can <a href="#">read more about the structure of the claims request in OpenID Connect section 5.5</a> ( <a href="https://openid.net/specs/openid-connect-core-1_0.html#ClaimsParameter">https://openid.net/specs/openid-connect-core-1_0.html#ClaimsParameter</a> ).
<b>max_age</b>	Optional	<b>max_age</b> is only available to services not on the GOV.UK domain and those handling particularly sensitive data. When the <b>max_age</b> parameter is included in your request, your user will be forced to re-authenticate if the time in seconds since authentication is greater than <b>max_age</b> . <b>max_age</b> must be set to zero or a positive integer.  You'll need to <a href="#">contact GOV.UK One Login support</a> ( <a href="https://www.sign-in.service.gov.uk/support">https://www.sign-in.service.gov.uk/support</a> ) to request to use <b>max_age</b> .
<b>code_challenge_length</b>	Optional	<b>code_challenge</b> is part of the Proof Key for Code Exchange (PKCE) protocol and helps protect against ‘Authorization Code’ interception attacks on authorisation requests. Your service generates the <b>code_challenge</b> by transforming a <b>code_verifier</b> using a <b>code_challenge_method</b> . This

parameter is required if your [client configuration enforces PKCE \(/configure-for-production/#use-the-table-to-understand-the-json-configuration\)](#).

You can [read more about PKCE in RFC 7636](#) (<https://datatracker.ietf.org/doc/html/rfc7636>).

`code_challenge_length_method` Optional

`code_challenge_method` specifies which [transformation method](#) (<https://datatracker.ietf.org/doc/html/rfc7636#section-4.2>) your service used to generate the `code_challenge`. If your request includes `code_challenge` you must include this field. This parameter is required if your [client configuration enforces PKCE \(/configure-for-production/#use-the-table-to-understand-the-json-configuration\)](#)

GOV.UK One Login only supports the `code_challenge_method S256`.

## Generate an authorisation code

If your user does not have an existing session they're signed in to when your service makes the request to the `/authorize` endpoint, the OIDC sign-in page will open. Your user can enter their details on this page to authenticate themselves.

If your user has an existing session, or after they authenticate, they will be redirected to the `redirect_uri` your service specified.

The authorisation code generated by your user's session can be used once and displays in the query string of the URL, for example:

```
HTTP/1.1 302 Found
```

```
Location: https://YOUR_REDIRECT_URI?code=AUTHORIZATION_CODE&state=xyzABC123
```

If your request included the `state` parameter, the URI will also include this parameter.

## Error handling for ‘Make a request to the `/authorize` endpoint’

You must check the HTTP return code from the `/authorize` request.

### HTTP 400 Bad Request

If your `GET` request to the `/authorize` endpoint produces a `Request is missing parameters` or `Invalid request` with `HTTP 400 (Bad Request)`, it might be because the parameters are not included correctly.

You should [check you have included the correct parameters \(/integrate-with-integration-environment/authenticate-your-user/#replace-the-placeholder-values-in-your-example\)](#), especially the `client_id`, `redirect_uri`, `response_type` and `scope` parameters.

## HTTP 302 Found

To understand more about what the error is, you can look in the response. Depending on the type of error you receive, the response may contain an `error` and an `error_description` which will provide you with information.

If there's an error in your request, you'll be redirected to the URI you specified for `redirect_uri` in the authorisation URL. You'll be able to see the error description tagged onto the end of the authorisation URL, for example:

```
HTTP/1.1 302 Found
Location: https://YOUR_REDIRECT_URI?error=invalid_request
&error_description=Unsupported%20response
&state=1234
```

Error	More information about your error
<code>unauthorized_client</code>	In rare circumstances, such as a security incident, One Login may prevent users from logging in to your service. If this happens, the error code <code>unauthorized_client</code> will be returned with the error description <code>client deactivated</code> . When your service receives this error, you must show the user a custom error page to explain that they cannot use your service at the moment and should try again later.
<code>request_is_missing_parameters</code>	The request has one or more of the following issues: <ul style="list-style-type: none"><li>missing a required parameter</li><li>includes an invalid parameter value</li><li>includes a parameter more than once</li><li>not in the correct format</li></ul> . You should <a href="#">check you have included the correct parameters (/integrate-with-integration-environment/authenticate-your-user/#replace-the-placeholder-values-in-your-example)</a> , especially the <code>client_id</code> , <code>redirect_uri</code> , <code>response_type</code> and <code>scope</code> parameters.
<code>invalid_request</code>	The request has one or more of the following issues: <ul style="list-style-type: none"><li>missing a required parameter</li><li>includes an invalid parameter value</li><li>includes a parameter more than once</li><li>not in the correct format</li></ul>

. You should [check you have included the correct parameters \(/integrate-with-integration-environment/authenticate-your-user/#replace-the-placeholder-values-in-your-example\)](#), especially the `client_id`, `redirect_uri`, `response_type` and `scope` parameters.

---

`invalid_request`  
-  
`Request vtr not valid`

You've requested single factor authentication and identity information. To make a successful identity request, you must request two-factor authentication and the identity level of confidence, for example `C1.Cm.P2`.

`invalid_scope`

The scope or scopes you have requested are invalid, unknown, or are not in the correct format.  
You can read more about scopes in [choosing which user attributes your service can request \(/before-integrating/choose-which-user-attributes-your-service-can-request/\)](#).

---

`unsupported_response_type`

Your service is not registered for the requested `response_type`.  
You must set the `response_type` to be code: `response_type=code`.

---

`server_error`

The GOV.UK One Login authentication server has experienced an internal server error.

---

`temporarily_unavailable`

If you're only making an authentication request (as opposed to requesting both authentication and identity), this error code means the GOV.UK One Login authentication server is temporarily unavailable, which might be caused by temporary overloading or planned maintenance.  
Make your request again in a few minutes.

If you're making an identity request and you get this error, it means the identity proving and verification does not currently have capacity for this request.

---

`access_denied`

GOV.UK One Login returns this error in 2 scenarios.

The first scenario is that the session in the user's browser is unavailable. This can happen when your user's cookies have been lost or your user changed browsers during the identity verification process. You should then [make another authentication and identity request \(<https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/authenticate-your-user/#make-a-request-for-authentication-and-identity>\)](#). You must help your user try again, for example by going back to the start of your authentication and identity verification process.

The second scenario is that the identity evidence your user provided has a lower score than the identity confidence specified in your request. As a result, GOV.UK One Login could not return the medium level of identity confidence ( P2 ) and instead returned a lower level of identity confidence.

If you're using return codes, you will not receive an error for this scenario. Find more information on [understanding the return codes claim \(/integrate-with-integration-environment/prove-users-identity/#understand-your-user-s-return-code-claim\)](#).

---

login_required	You have made a re-authentication request and a user is unable to authenticate themselves.
----------------	--

Reasons for this could be:

- the user has entered a different email address to the one stored
- the user has entered the wrong password too many times
- the user has entered the wrong one-time password (OTP) code too many times

## Make a token request

You need to exchange your [authorisation code \(/integrate-with-integration-environment/authenticate-your-user/#generate-an-authorisation-code\)](#) for tokens. You'll use these tokens to make a call to the [/userinfo](#) endpoint.

To exchange your authorisation code for tokens, you'll need to make a `POST` request to the `/token` endpoint using the client authentication method `private_key_jwt` or `client_secret_post` (only available for certain third-party platforms). There's further guidance on [using the correct token authentication method \(/before-integrating/use-correct-token-authentication-method/\)](#).

Before you can make a `POST` request, you need to:

1. Create a JWT assertion.
2. Include the JWT assertion in your `POST` request.

GOV.UK One Login will then authenticate your request by verifying the signature and payload of the JWT assertion. This authentication will generate a token response, which will include:

- an access token
- an ID token

## Create a JWT assertion

To create a JWT assertion, you need to:

1. Use the [key pair you generated \(/before-integrating/set-up-your-public-and-private-keys/#create-a-key-pair\)](#) earlier in the process.
2. Create a JWT.
3. Sign your JWT with the key you created - how you sign your JWT will vary depending on the language you're using.

## Create a JWT

To create a JWT assertion, you need to create a JWT which contains certain required claims. There's some optional claims you can choose to include or not include.

Claim	Required or recommended	Description
<code>aud</code>	Required	<code>aud</code> stands for ‘audience’. This identifies GOV.UK One Login’s authorisation server as an intended audience. This value should be the URL: <a href="https://oidc.integration.account.gov.uk/token">https://oidc.integration.account.gov.uk/token</a> .
<code>iss</code>	Required	<code>iss</code> stands for ‘issuer’. This claim should contain your <code>client_id</code> you got when you <a href="#">registered your service to use GOV.UK One Login (/before-integrating/register-and-manage-your-service/)</a> .
<code>sub</code>	Required	<code>sub</code> stands for ‘subject’. This claim should contain your <code>client_id</code> you got when you <a href="#">registered your service to use GOV.UK One Login (/before-integrating/register-and-manage-your-service/)</a> . There’s further guidance on how to use this value in the <a href="#">response to the /userinfo endpoint (<a href="https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/authenticate-your-user/#receive-response-for-retrieve-user-information">https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/authenticate-your-user/#receive-response-for-retrieve-user-information</a>)</a> .
<code>exp</code>	Required	<code>exp</code> stands for ‘expiration time’. This is the expiration time for this token, which must be an integer timestamp representing the number of seconds since the <a href="#">Unix Epoch (<a href="https://www.epochconverter.com/">https://www.epochconverter.com/</a>)</a> . This is the time after which you must not accept the JWT. We recommend an expiration after 5 minutes.

The current date and time must be before the expiration date and time listed in the `exp` claim.

---

<code>jti</code>	Required	<p><code>jti</code> stands for ‘JWT ID’. In this claim, you should include a unique identifier for the token. This unique identifier will prevent the token being reused as your application must only use these tokens once.</p>
<code>iat</code>	Recommended	<p><code>iat</code> stands for ‘issued at’. This identifies the time at which your application created the JWT. You can use this claim to understand the age of the JWT. This must appear as an integer timestamp representing the number of seconds since the <a href="https://www.epochconverter.com/">Unix Epoch</a> (<a href="https://www.epochconverter.com/">https://www.epochconverter.com/</a>).</p>

---

Your JWT body will look similar to this example:

```
{  
  "aud": "https://oidc.integration.account.gov.uk/token",  
  "iss": "229pcVGuHP1lXX37T7Wfbr5SIgm",  
  "sub": "229pcVGuHP1lXX37T7Wfbr5SIgm",  
  "exp": 1536165540,  
  "jti": "RANDOM_VALUE_JTI",  
  "iat": 1536132708  
}
```

Once you have created your JWT and signed your JWT with the key pair, you have created your JWT assertion.

## Make a POST request to the `/token` endpoint

Now you have generated your JWT assertion, you’re ready to make a `POST` request to the `/token` endpoint, for example:

```
POST /token HTTP/1.1  
Host: oidc.integration.account.gov.uk  
Content-Type: application/x-www-form-urlencoded  
User-Agent: my-platform/version (https://my-service-url.gov.uk)  
  
grant_type=authorization_code
```

```
&code=Spxl0BeZQQYbYS6WxSbIA  
&redirect_uri=https%3A%2F%2Fclient.example.org%2F  
&client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-type%3  
&client_assertion=eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIiLCJpc3Mi0iIiLCJhdWQi  
OiIiLCJqdGkiOiIifQ.r1Ylfhy6VNSlhIhW1N89F3WFIGuko2rvSRW04yK1BI
```

**!** This code example uses formatting that makes it easier to read. If you copy the example, you must make sure the request is:

- a continuous line of text separating each parameter with an ampersand (&)
- not split across multiple lines
- without any additional separators such as newline, commas or tabs

**!** GOV.UK One Login requires the [User-Agent header \(/before-integrating/set-user-agent-header\)](#) to be populated. If it absent or empty, your service will receive a 403 error

Parameter	Required or recommended	Description
grant_type	Required	You need to set the parameter to <code>authorization_code</code> .
redirect_uri	Required	You'll have specified your <code>redirect_uri</code> when you made the initial authorisation request.
client_assertion	Required	You'll include the JWT assertion you created in the payload when you make the <code>POST</code> request to the <code>/token</code> endpoint.
client_assertion_type	Required	When you're using <code>private_key_jwt</code> , you must set the value to <code>urn:ietf:params:oauth:client-assertion-type:jwt-bearer</code> .
code	Required	The code you received when you <a href="#">generated an authorisation code (/integrate-with-integration-environment/authenticate-your-user/#generate-an-authorisation-code)</a> .
code_verifier	Optional	You should only include this parameter if your original <code>/authorize</code> request includes the

`code_challenge` and `code_challenge_method` parameters.

## Receive response for ‘Make a token request’

If your token request is successful, the `/token` endpoint will return a response similar to this example:

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "access_token": "S1AV32hkKG",
  "token_type": "Bearer",
  "expires_in": 180,
  "id_token": "eyJhbGciOiJSUzI1NiIsImtpZCI6IjFlOWdkazcifQ.ewogImlzc
    yI6ICJodHRw0i8vc2VydmlvLmV4YW1wbGUuY29tIiwKICJzdWIiOiAiMjQ4Mjg"
}
```

You can use the following table to understand the response for ‘Make a token request’.

Parameter	Description
<code>access_token</code>	The access token value is an opaque access token which you can use with the <code>/userinfo</code> endpoint to return a user’s profile.
<code>token_type</code>	The token type value. GOV.UK One Login only supports the <a href="https://oauth.net/2/bearer-tokens/">bearer token</a> ( <a href="https://oauth.net/2/bearer-tokens/">https://oauth.net/2/bearer-tokens/</a> ).
<code>expires_in</code>	The length of time the token is valid for. This is displayed in seconds.
<code>id_token</code>	A signed JWT that contains basic attributes about the user.  By default, GOV.UK One Login signs this JWT using the <a href="#">ES256</a> algorithm.  If your service cannot support the <a href="#">ES256</a> algorithm (for example, some third-party tooling does not support <a href="#">ES256</a> ), GOV.UK One Login can sign the JWT using the <a href="#">RS256</a> algorithm. You’ll have specified whether your service can support <a href="#">ES256</a> when you <a href="#">registered your service to use GOV.UK One Login</a> ( <a href="#">/before-integrating/register-and-manage-your-service/</a> ).
	The public key used to verify this JWT is available from the <code>jwks_uri</code>

parameter found in the [discovery endpoint](#) (<https://oidc.integration.account.gov.uk/.well-known/openid-configuration>).

## Understand your ID token

The `id_token` parameter in the response for ‘Make a token request’ contains the following claims:

```
{  
  "at_hash": "ZDevf74CkYWNPa8qmflQyA",  
  "sub": "urn:fdc:gov.uk:2022:VtcZjnU4Sif2oyJZola30kN0e3Jeku1cIMN38rFlhU4",  
  "aud": "YOUR_CLIENT_ID",  
  "iss": "https://oidc.integration.account.gov.uk/",  
  "vot": "Cl.Cm",  
  "exp": 1704894526,  
  "iat": 1704894406,  
  "nonce": "lZk16Vmu8-h7r8L8bFFiHJxpC3L73UBpf68WC1Qoqg",  
  "vtm": "https://oidc.integration.account.gov.uk/trustmark",  
  "sid": "dX5xv0XgHh6yfD1xy-ss_1EDK0I"  
  "auth_time": 1704894300  
}
```

You can use the following table to understand the ID token’s claims.

### Claim    Description

<code>at_ha sh</code>	<code>at_hash</code> stands for ‘access token hash’. You use <code>at_hash</code> to validate your access token. This is not mandatory. There is further <a href="#">guidance on <code>at_hash</code> in the Open ID Connect specification</a> ( <a href="https://openid.net/specs/openid-connect-core-1_0.html#CodeIDToken">https://openid.net/specs/openid-connect-core-1_0.html#CodeIDToken</a> ).
<code>sub</code>	<code>sub</code> stands for the subject identifier or the unique ID of a user.
<code>aud</code>	<code>aud</code> stands for the audience, which will be the <code>client_id</code> you received when you <a href="#">registered your service to use GOV.UK One Login</a> ( <a href="/before-integrating/register-and-manage-your-service/">/before-integrating/register-and-manage-your-service/</a> ).
<code>iss</code>	<code>iss</code> stands for the GOV.UK One Login OpenID Provider’s Issue identifier as specified in the <a href="#">discovery endpoint</a> ( <a href="https://oidc.integration.account.gov.uk/.well-known/openid-configuration">https://oidc.integration.account.gov.uk/.well-known/openid-configuration</a> ).

---

vot	vot stands for ‘Vector of Trust’. Check the vot matches the authentication protection level you requested in your authorise request. The vot claim will only contain the credential trust level and not the level of confidence, even if you make an identity request.
exp	exp stands for ‘expiration time’. This is the expiration time for this token, which will be an integer timestamp representing the number of seconds since the <a href="https://www.epochconverter.com/">Unix Epoch (https://www.epochconverter.com/)</a> .
iat	iat stands for ‘issued at’. This identifies the time at which GOV.UK One Login created the JWT. You can use this claim to understand the age of the JWT. This will appear as an integer timestamp representing the number of seconds since the <a href="https://www.epochconverter.com/">Unix Epoch (https://www.epochconverter.com/)</a> .
nonce	The nonce value your application provided when you made the request to the /authorize endpoint.
vtm	vtm stands for ‘vector trust mark’. This is an HTTPS URL which lists the range of values GOV.UK One Login accepts and provides.
sid	sid stands for ‘session identifier’. This uniquely identifies the user’s journey within GOV.UK One Login.
auth_time	auth_time is the time at which your user last authenticated. This will be an integer timestamp representing the number of seconds since the <a href="https://www.epochconverter.com/">Unix Epoch (https://www.epochconverter.com/)</a> .

---

Now you’ve understood what’s in your ID token, you’ll need to validate it.

## Validate your ID token

 **You must perform all of the validation described below, or your integration may not be secure**

1. If you’re using a library, check whether your library has support for validating ID tokens.
2. The value of iss must exactly match the Issuer Identifier as specified in GOV.UK One Login’s [discovery endpoint \(https://oidc.integration.account.gov.uk/.well-known/openid-configuration\)](https://oidc.integration.account.gov.uk/.well-known/openid-configuration).
3. The aud claim must contain your client ID you received when you [registered your service to use GOV.UK One Login \(/before-integrating/register-and-manage-your-service/\)](#).
4. You must validate the signature according to the [JSON Web Signature Specification \(https://datatracker.ietf.org/doc/html/rfc7515\)](https://datatracker.ietf.org/doc/html/rfc7515). You must first validate that the JWT alg header matches (<https://datatracker.ietf.org/doc/html/rfc8725#section-3.1>) what was returned

from the `jwks_uri`. Then you can use the value of the JWT `alg` header parameter to validate the ID token. Your application must use the keys provided by the [discovery endpoint](https://oidc.integration.account.gov.uk/.well-known/openid-configuration) (<https://oidc.integration.account.gov.uk/.well-known/openid-configuration>).

5. Check the current time is before the time in the `exp` claim.
6. Check the current time is after the time in the `iat` claim.
7. If you set a `nonce` value in the request to the `/authorize` endpoint, check this matches the `nonce` value in the ID token.
8. The `vot` claim must contain the credential trust level you asked for in the request to the `/authorize` endpoint. The `vot` claim will only contain the credential trust level, not the level of confidence, even if you make an identity request. For example, if you set the `vtr` parameter to `C1.Cm.P2`, you must ensure the `vot` claim is equal to `C1.Cm`.
9. If you included `max_age` in the request to the `/authorize` endpoint, you must validate that `auth_time` is greater than or equal to the current time subtract the value of `max_age`. If false, you should reject the ID token and redirect the user to re-authenticate, by sending a new authorisation request including `max_age`.

## Error handling for ‘Make a token request’

To understand more about what the error is, you can look in the response. Depending on the type of error you receive, the response may contain an `error` and an `error_description` which will provide you with information.

If the token request is invalid or unauthorised, you’ll receive an error response with the `Content-Type` of `application/json`, for example:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
{
  "error": "invalid_request"
  "error_description": "invalid scope"
}
```

Error	More information about your error
<code>invalid_request</code>	The request is missing a parameter so the server cannot proceed with the request. This error may also be returned if the request includes an unsupported parameter or repeats a parameter.  Review your parameters and check they are supported and not repeated.
<code>invalid_client</code>	Client authentication failed, which could be caused by the request containing an invalid <code>client_id</code> or an issue in validating the signature of the <code>client_assertion</code> .

To resolve, check:

- your `client_id` matches the `client_id` you received when you [registered your service to use GOV.UK One Login \(/before-integrating/register-and-manage-your-service/\)](#)
  - you have signed your `client_assertion` JWT with the private key generated when you [registered your service to use GOV.UK One Login \(/before-integrating/register-and-manage-your-service/\)](#)
  - your service uses a [key signing algorithm which GOV.UK One Login supports \(https://oidc.account.gov.uk/.well-known/openid-configuration\)](#)
- 

`invalid_grant`

The request has one or more of the following issues:

- the authorisation code is invalid or expired
  - the redirect URL given in the authorisation request does not match the URL provided in this access token request
  - the authorisation request included Proof Key for Code Exchange (PKCE) parameters, and the `code_verifier` is missing or invalid
- 

`unauthorized_client`

The application is successfully authenticated, but it's not registered to use the requested [grant type \(https://oauth.net/2/grant-types/\)](#).

`unsupported_grant_type`

The grant type is not supported by the server.

## Retrieve user information

You can retrieve information about your users by making a request to the `/userinfo` endpoint.

Make the request to the `/userinfo` endpoint using the access token you received when making a token request. Using the authorisation header field, send the access token as a [bearer token \(https://oauth.net/2/bearer-tokens/\)](#). You'll receive a JSON object which contains a collection of name and value pairs.

An example request to the `/userinfo` endpoint would look similar to this example:

```
GET /userinfo HTTP/1.1
Host: oidc.integration.account.gov.uk
Authorization: Bearer <access_token>
User-Agent: my-platform/version (https://my-service-url.gov.uk)
```

**!** GOV.UK One Login requires the [User-Agent header \(/before-integrating/set-user-agent-header/\)](#) to be populated. If it absent or empty, your service will receive a 403 error

## Receive response for ‘Retrieve user information’

The response you’ll get after making a request to the `/userinfo` endpoint will be a JSON object containing user attributes.

If you included all the scopes when you were [choosing which user attributes your service can request \(/before-integrating/choose-which-user-attributes-your-service-can-request/\)](#) and made a request to the `/userinfo` endpoint, the response would look similar to this:

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "sub": "urn:fdc:gov.uk:2022:56P4CMsGh_02Y0lWpd8PA0I-2sV1B2nsNU7mcLZYhYw=",
  "email": "test@example.com",
  "email_verified": true,
  "phone_number": "+441406946277",
  "phone_number_verified": true
}
```

If you included a [level of identity confidence \(/before-integrating/choose-the-level-of-identity-confidence/\)](#) when you made a request to the `/userinfo` endpoint, you’ll also see identity attributes in the response. You can read more about [how to prove your user’s identity](#).

Claim returned	Description
<code>sub</code>	<p>The subject identifier (<code>sub</code>) is the unique ID for a user. This will not change unless your user deletes their GOV.UK One Login and sets it up again.</p> <p>Do not use the <code>sub</code> as the primary identifier for your user.</p> <p>Instead, generate your own unique value for your user within your service and map this against the GOV.UK One Login <code>sub</code>.</p> <p>Mapping the <code>sub</code> makes account recovery easier. For example, if a user deletes their GOV.UK One Login, you can re-map the user’s new <code>sub</code> to your service without creating a new primary identifier for your user.</p>

**email** The email address your user entered when they registered their GOV.UK One Login.

Do not:

- use `email` as the primary identifier for your user (the `email` claim can change or an end user can lose access to it which makes it unreliable as a unique identifier)
- ask your user to create a GOV.UK One Login with a specific email address, for example, a university email – if you need this, you'll need to build additional functionality to verify it yourself
- ask your user to change the email address they use for their GOV.UK One Login

---

**email\_verified** This means the email was verified using a one-time code when the user created their account. This is always `true`.

**phone\_number** This is the phone number your user entered when they registered their GOV.UK One Login. This will not appear if the user used an authenticator app for their two-factor authentication.

This will return in the E.164 format with no spaces for both UK and international phone numbers: `+\{country-code}\Number`.

---

**phone\_number\_verified** This will be returned as:

- `true` when the user has selected the text message option for receiving a security code
- `false` when the user has selected the authenticator app option for receiving a security code

---

**walletSubjectId** This will be returned in the format:  
`urn:fdc:wallet.account.gov.uk:2024:3c_jJtXcLttICSNrkW7M1v02_w-SMDm2nrHsZpWQQ9`

where the part after `urn:fdc:` is [Base 64 Encoding with URL and Filename Safe Alphabet](#) (<https://datatracker.ietf.org/doc/html/rfc4648#section-5>) of the output from a SHA256 hash function.

**walletSubjectId** is a pairwise identifier that GOV.UK Wallet uses when it issues a credential. By comparing the returned value with the value GOV.UK Wallet submits when requesting a credential, you can be sure that the user logged into your service and GOV.UK Wallet are the same user.

You must include this scope if you plan to [onboard with GOV.UK Wallet](#) (<https://docs.wallet.service.gov.uk/before-integrating.html#onboard-with-gov-uk-one-login>) after you have onboarded with GOV.UK One Login.

# Error handling for ‘Retrieve user information’

To understand more about what the error is, you can look in the response. Depending on the type of error you receive, the response may contain an `error` and an `error_description` which will provide you with information.

When a request fails, the `/userinfo` endpoint will respond with:

- an HTTP status code (usually 401 or 403)
- an error code (usually `error` parameter and an `error_description`) included in the response

An error response for the `/userinfo` endpoint would look similar to this example:

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Bearer error="invalid_token",
error_description="The Access Token expired"
```

Error	More information about your error
<code>invalid_token</code>	GOV.UK One Login denied your request as you have an invalid or missing bearer access token.  To proceed, you must use the authorisation header field to send the token as a <a href="https://oauth.net/2/bearer-tokens/">bearer token (https://oauth.net/2/bearer-tokens/)</a> .

Once you’ve authenticated your user, you can continue with [proving your user’s identity \(/integrate-with-integration-environment/prove-users-identity/\)](#).

If you’re only authenticating your users, skip the next section and move onto [managing your users’ sessions \(https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/managing-your-users-sessions/#managing-your-users-39-sessions\)](#).

This page was last reviewed on 12 February 2025.



## Accessibility

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Helping your users after their in person identity checks

-  If you need identity verification, we strongly recommended that you create a landing page / [LandingPageUrl1](#) to support your users after their in-person identity checks. This is also known as the face-to-face journey. There is further [guidance on creating and testing a landing page \(https://github.com/govuk-one-login/onboarding-examples/tree/main/tools/f2f-test\)](#). If you do not provide a landing page, your user will not be returned to your service after they have completed their in-person identity checks. If your service is authentication-only, you do not need to do this.

In some scenarios, your user may have to prove their identity ‘face-to-face’, for example after using the Post Office or if they have a European driving licence.

1. Trained identity staff process the user’s documents and take their photo to compare the user’s claimed identity to their documentation.
2. The identity staff do not tell the user the result of their identity check immediately. Instead, your user will receive an email with the subject **Sign in to view the result of your identity check**, which can take up to 24 hours to arrive.
3. The user selects the link in the email’s body (this contains an identifier which links them to your service), signs in to GOV.UK One Login and is automatically redirected to your service’s landing page.

Your user has 16 days to complete their in-person identity check, starting from when they initiate the journey. The in-person journey is only complete when your user visits GOV.UK One Login after a success or failure of the in-person check. Your user can do this by using their unique email link or trying to access your service or another service connected to GOV.UK One Login.

If the user does not complete the in-person identity check within 16 days, they will not be able to complete it at this time. The user will then have to repeat the identity check process.

Currently, the only way to cancel an in-person identity check is for the user to contact GOV.UK One Login directly.

# Understand what your landing page needs to do

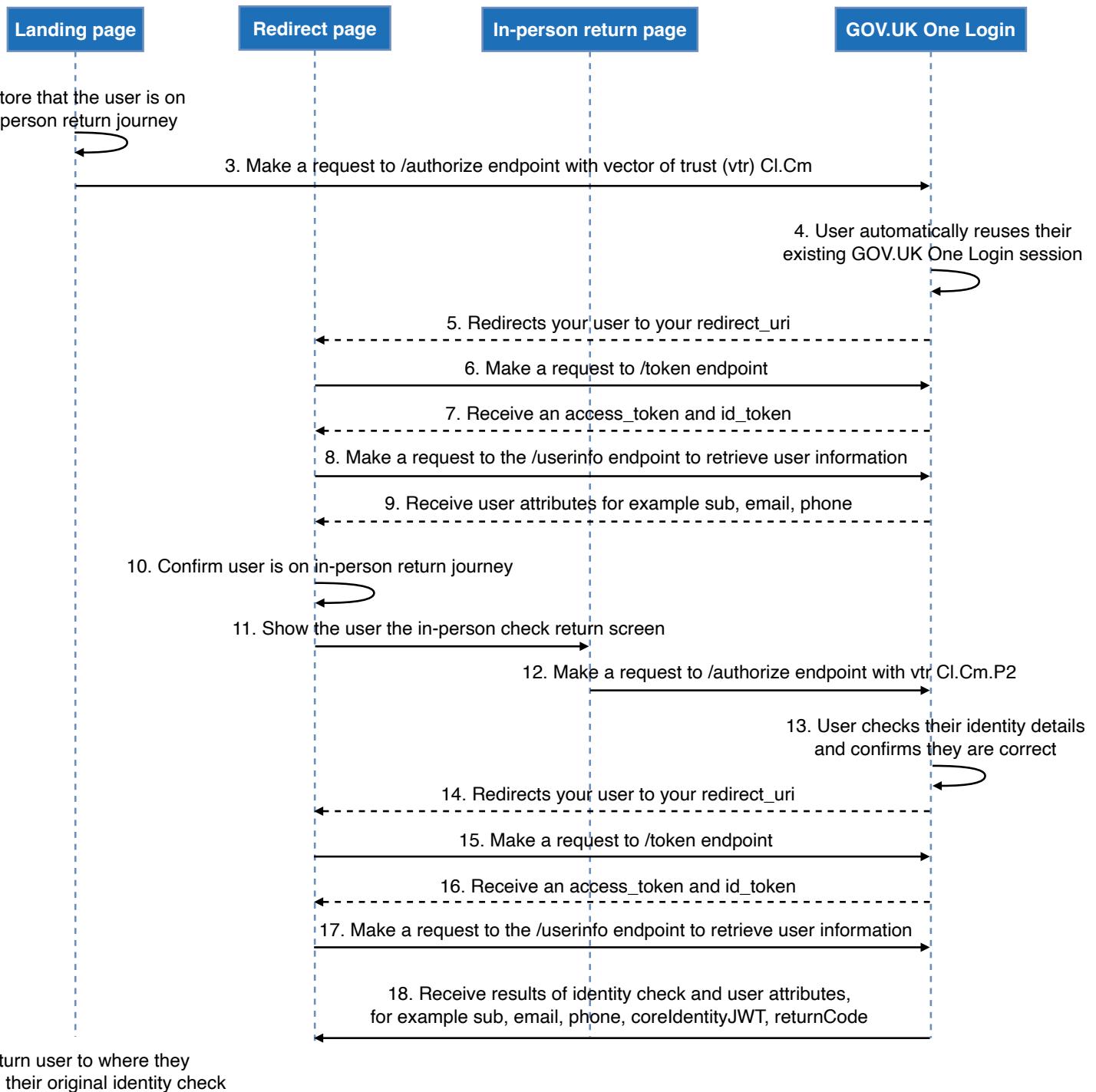
This landing page will:

- authenticate a user in your service
- help your user to see the status of their recent identity check
- continue your user's sign-up process for your service

Before you start, you must have followed the recommendation to [split out authentication and identity requests \(/integrate-with-integration-environment/authenticate-your-user/#make-a-request-for-authentication-and-identity\)](#).

We recommend your user returns to the point where they left to do the in-person check rather than starting over again. You can do this if you stored the user's `sub` when the user initially authenticated with your service.

1. User with existing session lands on LandingPageUrl



1. Your user, having used their unique link from the **Sign in to view the result of your identity check** email, logged in to GOV.UK One Login and viewed their identity check status, has selected to return to your service, and lands on the **LandingPageUrl** that you gave when you [configured your service for production \(/configure-for-production/\)](#).

2. This is where the in-person journey differs from a normal GOV.UK One Login flow. Your service's landing page needs to store that this user is on an in-person return journey. How you do this will depend on how you [manage your users' sessions \(/integrate-with-integration-environment/managing-your-users-sessions/#managing-your-users-39-sessions\)](#). Your user should only end up on this page if they have selected the email link from the in-person journey.

3. Your service makes an authentication request to the `/authorize` endpoint with the vector of trust (vtr) `C1.Cm` (a medium level of authentication). There's further [guidance on making an authentication request \(/integrate-with-integration-environment/authenticate-your-user/#make-a-request-for-authentication\)](#).
4. Your user automatically reuses their existing GOV.UK One Login session because they have already logged in from the email link. The user will not need to re-enter their username and password.
5. GOV.UK One Login redirects your user to your `redirect_uri`.
6. Your service makes a [token request \(<https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/authenticate-your-user/#make-a-token-request>\)](#) to the `/token` endpoint.
7. Your service receives an [ID token and access token \(<https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/authenticate-your-user/#receive-response-for-make-a-token-request>\)](#) in the response.
8. Your service makes a request to the `/userinfo` endpoint to retrieve user information.
9. Your service receives a response containing the user attributes `sub`, `email` and `phone`.
10. Your service confirms that your user is on an in-person return journey (you'll have stored that this user is on an in-person return journey in step 2).
11. This is where the in-person journey differs from a normal GOV.UK One Login flow. Your service shows the user the 'in-person check return screen' which prompts the user to continue their journey for retrieving the result of their identity check. This will be the first page your service shows the user.
12. Your service makes a request for both authentication and identity to the `/authorize` endpoint with the vector of trust `C1.Cm.P2` (medium authentication and medium level of identity confidence). There's further [guidance on making an authentication and identity request \(/integrate-with-integration-environment/authenticate-your-user/#make-a-request-for-authentication-and-identity\)](#).
13. Your user checks the identity details GOV.UK One Login has stored for them and confirms they are correct.
14. GOV.UK One Login redirects your user to your `redirect_uri`. Your service makes a [token request \(<https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/authenticate-your-user/#make-a-token-request>\)](#) to the `/token` endpoint.
15. Your service receives an [ID token and access token \(<https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/authenticate-your-user/#receive-response-for-make-a-token-request>\)](#) in the response.
16. Your service makes a request to the `/userinfo` endpoint to retrieve user information.
17. Your service receives a response containing user attributes (`sub`, `email`, `phone` and whichever claims your service requested, for example `coreIdentityJWT`) and the results of the identity check.
18. Your service returns your user to where they started their original identity check.

## Test your LandingPageUrl

You cannot directly test the in-person ‘face-to-face’ journey. This is because your user will access your service’s [LandingPageUrl](#) from their unique link in the **Sign in to view the result of your identity check** email.

However, you can [use the f2f-test tool to test a simulated return from the face-to-face journey \(<https://github.com/govuk-one-login/onboarding-examples/blob/main/tools/f2f-test/README.md>\)](#).

This page was last reviewed on 4 September 2025.

[View source](#) [Report problem](#) [GitHub Repo](#)



## Accessibility

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Managing your users' sessions

GOV.UK One Login's session timeout duration is 1 hour. The 1 hour timeout starts when your user last interacts with GOV.UK One Login, not 1 hour from when they start their journey. You have different methods to manage a user's session depending on the session timeout duration of your service. If this duration is:

- less than 1 hour: there's [guidance on managing your users' sessions if using a session expiry below 1 hour \(/integrate-with-integration-environment/managing-your-users-sessions/#managing-user-sessions-if-your-service-session-is-less-than-1-hour\)](#)
- 1 hour: both your session and GOV.UK One Login's expire after 1 hour and you send a request to the [/logout endpoint](#) to log your users out
- more than 1 hour: GOV.UK One Login's session will expire before your session, so your user has to reauthenticate themselves if they need to log in to another service after this time

 **All services should build functionality to log a user out. However, if your session timeout duration is less than 1 hour, you must build functionality for your users to log themselves out of your service and GOV.UK One Login.**

## Re-authenticating your users

You may want to make sure your user is required to sign in interactively even when they have an existing GOV.UK One Login session.

GOV.UK One Login only supports re-authentication requests using a populated ID token and [prompt=login](#) for JWT-secured OAuth 2.0 authorisation requests (JARs).

GOV.UK One Login allows retries for each credential a user gets wrong. If a user exceeds the maximum retries allowed before the retry count expires then GOV.UK One Login logs the user out and will send the [login\\_required error \(/integrate-with-integration-environment/authenticate-your-user/#error-handling-for-make-a-request-to-the-authorize-endpoint\)](#) to your service. You must handle this error and terminate the user's session in your service for security.

# Managing user sessions if your service session is less than 1 hour

We advise that your service has either the same or a longer session expiry than GOV.UK One Login.

If your service has a session expiry shorter than 1 hour and your user's session in your service has expired, GOV.UK One Login will automatically log your user back in if they return to your service. Your user will not have to re-enter their username and password and there is no disruption to their journey. This also applies if your user is using another service integrated with GOV.UK One Login.

## Build functionality to log your user out

 **All services should build functionality to log a user out. However, if your session timeout duration is less than 1 hour, you must build functionality for your users to log themselves out of your service and GOV.UK One Login.**

You must do this because the GOV.UK One Login session cookie is persistent and remains valid even if the device or browser is closed. If your users share devices, for example in a workplace or family laptop, there could be a risk of users accidentally sharing sessions if they cannot log themselves out.

You have different options to build functionality to log your users out:

- use the [GOV.UK One Login service header](https://www.sign-in.service.gov.uk/documentation/design-recommendations/let-users-navigate-sign-out) (<https://www.sign-in.service.gov.uk/documentation/design-recommendations/let-users-navigate-sign-out>) which contains a built-in Sign out button
- if your application ends in a user selecting Submit, code the submit button to automatically log the user out
- build an auto-logout after a period of inactivity from a user
- include a logout button

All of these options must send a logout query to the `/logout` endpoint to end the user's session.

## Log your user out of GOV.UK One Login

To log users out of GOV.UK One Login, you need to call the `/logout` endpoint.

You can also [request logout notifications from GOV.UK One Login](#) ([/integrate-with-integration-environment/managing-your-users-sessions/#request-logout-notifications-from-gov-uk-one-login](#)).

## Make a request to ‘Log your user out of GOV.UK One Login’

You must set up the functionality to log users out of a GOV.UK One Login session.

1. Log your user out of using your application - the way you do this will depend on how you have built your service.
2. In the user’s browser, make a `GET` request to GOV.UK One Login’s `/logout` endpoint to end your user’s session.

```
HTTP/1.1 GET
Location: oidc.integration.account.gov.uk?
id_token_hint=eyJraWQiOiIxZTlnZGs3I...
&post_logout_redirect_uri=http://example-service.com/my-logout-url
&state=sadk8d4--lda%d
```

 **This code example uses formatting that makes it easier to read. If you copy the example, you must make sure the request is:**

- **a continuous line of text separating each parameter with an ampersand (&)**
- **not split across multiple lines**
- **without any additional separators such as newline, commas or tabs**

Parameter	Required, recommended or optional	Description
<code>id_token_hint</code>	Recommended - however, if you use <code>post_logout_redirect_uri</code> , this parameter is required	This is the ID token GOV.UK One Login previously issued when you made a request to the <code>/token</code> endpoint for your user’s current session.
<code>post_logout_redirect_uri</code>	Optional - however, if you do not specify this parameter, the endpoint redirects your user to the default logout page for GOV.UK One Login	You can only use this parameter if you have specified an <code>id_token_hint</code> . This parameter is the URL you want to redirect your users to after you have logged them out. The <code>post_logout_redirect_uri</code>

must match the logout URI you specified when you registered your service to use GOV.UK One Login.

---

state	Optional	You can use this query parameter to maintain state between the logout request and your user being redirected to the <a href="#">post_logout_redirect_uri</a> .
-------	----------	--

---

## Receive response for ‘Log your user out of GOV.UK One Login’

After you have made your [GET](#) request to GOV.UK One Login’s [/logout](#) endpoint, you should receive a response similar to this:

```
HTTP 1.1 302 Found
```

```
Location: https://example-service.com/my-logout-url&state=sadk8d4--1da%
```

You have now logged your user out of GOV.UK One Login and terminated all their sessions.

## Request logout notifications from GOV.UK One Login

GOV.UK One Login can use a [POST](#) request to notify you when a user who has previously logged into your service using GOV.UK One Login has logged out.

These notifications are optional, but we recommend supporting them, otherwise your service will not know if your user has logged out.

You can request to receive logout notifications by providing a [back\\_channel\\_logout\\_uri](#) when you [register your service to use GOV.UK One Login](#) ([/before-integrating/register-and-manage-your-service/](#)).

You can only supply one [back\\_channel\\_logout\\_uri](#) per client.

When you receive a logout notification for an end user, you must close all the sessions you hold for that user in your service.

The logout notifications follow the [OIDC back-channel logout specification](#) ([https://openid.net/specs/openid-connect-backchannel-1\\_0.html#Backchannel](https://openid.net/specs/openid-connect-backchannel-1_0.html#Backchannel)).

There's an [example implementation of handling a back-channel logout notification](https://github.com/govuk-one-login/relying-party-stub/blob/main/src/main/java/uk/gov/di/handlers/BackChannelLogoutHandler.java) (<https://github.com/govuk-one-login/relying-party-stub/blob/main/src/main/java/uk/gov/di/handlers/BackChannelLogoutHandler.java>).

You must make sure your `back_channel_logout_uri` can accept `POST` requests with a `Content-Type` of `application/x-www-form-urlencoded` from GOV.UK One Login.

The `back_channel_logout_uri` must be available using the internet. Using `localhost` will not work.

GOV.UK One Login will send a `POST` request to your `back_channel_logout_uri` when a user who has logged into your service using GOV.UK One Login has logged out. The `POST` body will contain a `logout_token`, which will be a signed JSON web token (JWT).

Here's an example of a decoded back-channel logout token:

```
{  
    "kid": "644af598b780f54106c2465489765230c4f8373f35f32e18e3e40cc7acff6",  
    "alg": "ES256"  
}.{  
    "iss": "https://oidc.integration.account.gov.uk/",  
    "sub": "urn:fdc:gov.uk:2022:56P4CMsGh_02Y0lWpd8PA0I-2sV1B2nsNU7mcLZYhYw=",  
    "aud": "YOUR_CLIENT_ID",  
    "iat": 1713185467,  
    "exp": 1713185587,  
    "jti": "30642c87-6167-413f-8ace-f1643c59e398",  
    "events": {  
        "http://schemas.openid.net/event/backchannel-logout": {}  
    }  
}
```

As an end user might have multiple sessions with your service, you may receive multiple logout notifications for the same user.

## Validate your logout token

Once you've received a `POST` request to your `back_channel_logout_uri`, you must validate the JWT signature and logout token payload.

1. Validate that the JWT `kid` claim in the logout token header exists in the JWKS (JSON web key set) returned by the [/jwks endpoint](https://oidc.integration.account.gov.uk/.well-known/jwks.json) (<https://oidc.integration.account.gov.uk/.well-known/jwks.json>).
2. Check the JWT `alg` header matches the value for the key you are using.

3. Use the key to validate the signature on the logout token according to the [JSON Web Signature Specification](https://datatracker.ietf.org/doc/html/rfc7515) (<https://datatracker.ietf.org/doc/html/rfc7515>).
4. Check the value of `iss` (issuer) matches the Issuer Identifier specified in GOV.UK One Login's [discovery endpoint](https://oidc.integration.account.gov.uk/.well-known/openid-configuration) (<https://oidc.integration.account.gov.uk/.well-known/openid-configuration>).
5. Check the `aud` (audience) claim is the same client ID you received when you [registered your service to use GOV.UK One Login](#) ([/before-integrating/register-and-manage-your-service/](#)).
6. Check the `iat` (issued at) claim is in the past.
7. Check the `exp` (expiry) claim is in the future.
8. Check the logout token contains a `sub` (subject identifier) claim, otherwise known as the unique ID of a user.
9. Check the logout token contains an `events` claim, which should be a JSON object with a single key: `http://schemas.openid.net/event/backchannel-logout` – the value for the key should be an empty object.
10. Check your service has not received another logout token with the same `jti` claim in the last 3 minutes.

If all the validation steps pass, you should close all the sessions for the user whose subject ID matches the `sub` claim in the payload.

## Respond to the back-channel logout request

You must respond to the back-channel logout HTTP request with an `HTTP 200 OK` response code. This will indicate whether you have received the logout request.

This page was last reviewed on 2 May 2024.

[View source](#) [Report problem](#) [GitHub Repo](#)



[Accessibility](#)



[Table of contents](#)

# Prove your user's identity

You must have authenticated your users before you can prove their identity.

If you [requested identity proving \(/before-integrating/choose-the-level-of-identity-confidence/\)](#), when you [retrieve user information with /userinfo \(/integrate-with-integration-environment/authenticate-your-user/#retrieve-user-information\)](#), you'll receive a response containing additional claims (user attributes). You may receive different claims, depending on how your user proved their identity.

Your service's needs will determine how you process the other claims that GOV.UK One Login provides about your user. You'll probably need to match against information held by your service or organisations you work with.

Most claims are represented by JSON objects. The [core identity claim](#) is a JSON web token (JWT) protected by an electronic signature for additional security.

You'll receive a response from [/userinfo](#) that will look similar to this example:

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

```
"sub": "urn:fdc:gov.uk:2022:56P4CMsGh_02Y01Wpd8PA0I-2sV1B2nsNU7mcLZYhYw=",
"email": "test@example.com",
"email_verified": true,
"phone": "+441406946277",
"phone_verified": true,
"https://vocab.account.gov.uk/v1/coreIdentityJWT": <JWT>,
"https://vocab.account.gov.uk/v1/address": [
{
  "uprn": "10022812929",
  "subBuildingName": "FLAT 5",
  "buildingName": "WEST LEA",
  "buildingNumber": "16",
  "dependentStreetName": "KINGS PARK",
  "streetName": "HIGH STREET",
```

```
"doubleDependentAddressLocality": "EREWASH",
"dependentAddressLocality": "LONG EATON",
"addressLocality": "GREAT MISSENDEN",
"postalCode": "HP16 0AL",
"addressCountry": "GB",
"validFrom": "2022-01-01"
},
{
"uprn": "10002345923",
"buildingName": "SAWLEY MARINA",
"streetName": "INGWORTH ROAD",
"dependentAddressLocality": "LONG EATON",
"addressLocality": "NOTTINGHAM",
"postalCode": "BH12 1JY",
"addressCountry": "GB",
"validUntil": "2022-01-01"
}
],
"https://vocab.account.gov.uk/v1/drivingPermit": [
{
"expiryDate": "2023-01-18",
"issueNumber": "5",
"issuedBy": "DVLA",
"personalNumber": "DOE99802085J99FG"
}
],
"https://vocab.account.gov.uk/v1/passport": [
{
"documentNumber": "1223456",
"icaoIssuerCode": "GBR",
"expiryDate": "2032-02-02"
}
]
}
```

## Understand your user's core identity claim

The <https://vocab.account.gov.uk/v1/coreIdentityJWT> property in the [/userinfo](#) response is the core identity claim, which is a JWT representing core identity attributes.

The following are core identity attributes:

- your user's name
- your user's date of birth
- the level of identity confidence GOV.UK One Login has reached

The core identity is valid for 30 minutes, starting when it is issued. Do not store the [coreIdentityJWT](#) in its raw encoded or decoded forms.

If your service persists the data inside the core identity, you should extract the name and date of birth and store those.

 If the <https://vocab.account.gov.uk/v1/coreIdentityJWT> property is not present, then GOV.UK One Login was not able to prove your user's identity.

You'll need a public key to validate this JWT. You can download a Decentralized Identifiers (DID) document containing the current JSON Web Key (JWK) public key – there's further [guidance on validating the core identity claim JWT using a public key](#) ([/integrate-with-integration-environment/prove-users-identity/#validate-the-core-identity-claim-jwt-using-a-public-key](#)).

The JWT contains the following claims:

```
{  
  "sub": "urn:fdc:gov.uk:2022:56P4CMsGh_02Y01Wpd8PA0I-2sV1B2nsNU7mcLZYhYw=",  
  "iss": "https://identity.integration.account.gov.uk/",  
  "aud": "YOUR_CLIENT_ID",  
  "nbf": 1541493724,  
  "iat": 1541493724,  
  "exp": 1573029723,  
  "vot": "P2",  
  "vtm": "https://oidc.integration.account.gov.uk/trustmark",  
  "vc": {  
    "type": [  
      "VerifiableCredential",  
      "VerifiableIdentityCredential"  
    ],  
    "credentialSubject": {  
      "name": [  
        "John Doe"  
      ]  
    }  
  }  
}
```

```
{  
    "validFrom": "2020-03-01",  
    "nameParts": [  
        {  
            "value": "Alice",  
            "type": "GivenName"  
        },  
        {  
            "value": "Jane",  
            "type": "GivenName"  
        },  
        {  
            "value": "Laura",  
            "type": "GivenName"  
        },  
        {  
            "value": "Doe",  
            "type": "FamilyName"  
        }  
    ]  
},  
{  
    "validUntil": "2020-03-01",  
    "nameParts": [  
        {  
            "value": "Alice",  
            "type": "GivenName"  
        },  
        {  
            "value": "Jane",  
            "type": "GivenName"  
        },  
        {  
            "value": "Laura",  
            "type": "GivenName"  
        },  
        {  
            "value": "O'Donnell",  
            "type": "FamilyName"  
        }  
    ]  
}
```

```
        ],
      ],
      "birthDate": [
        {
          "value": "1970-01-01"
        }
      ]
    }
  }
}
```

The `vc` claim in the JWT is a [verifiable credential \(VC\)](https://www.w3.org/TR/vc-data-model/) (<https://www.w3.org/TR/vc-data-model/>). Claims about your user are contained in the `credentialSubject` JSON object.

## Validate the core identity claim JWT using a public key

To validate the core identity claim JWT, you must use a public key. GOV.UK One Login publishes the public keys in a [Decentralized Identifier \(DID\) document](https://www.w3.org/TR/did-core/) (<https://www.w3.org/TR/did-core/>).

 **GOV.UK One Login regularly rotates its public keys. You must [read the guidance on understanding GOV.UK One Login's key rotation](#) ([/integrate-with-integration-environment/prove-users-identity/#understand-the-core-identity-signing-key-rotations](#)) to make sure your application continues to work as expected.**

This is an example of a web DID document published by GOV.UK One Login:

```
{
  "@context": [
    "https://www.w3.org/ns/did/v1",
    "https://w3id.org/security/jwk/v1"
  ],
  "id": "did:web:identity.account.gov.uk",
  "assertionMethod": [
    {
      "id": "did:web:identity.account.gov.uk#b7863b6926193d93b48808cbabcbc8a",
      "type": "JsonWebKey",
      "controller": "did:web:identity.account.gov.uk",
      "publicKeyJwk": {
        "kty": "EC",
        "crv": "P-256",
        "x": "A long string of hex digits representing the X coordinate of the public key point",
        "y": "A long string of hex digits representing the Y coordinate of the public key point",
        "d": null
      }
    }
  ]
}
```

```
        "crv": "P-256",
        "x": "QrP65yghuglwPkEl11oMaabr4WqAMjuvztBYb7T4Ipo",
        "y": "CSQNybYbCZLl-Xr10A3pcxjC6qZrG7JPqwXgo-9fHLM"
    }
}
]
}
```

## Use the `kid` (key ID) to see which public key signed the JWT

When validating a JWT, the JWT header will include the `kid` (key ID). This will be either `did:web:identity.integration.account.gov.uk#{UNIQUE_KEY_IDENTIFIER}` for the integration environment, or `did:web:identity.account.gov.uk#{UNIQUE_KEY_IDENTIFIER}` for the production environment.

GOV.UK One Login uses a simplified version of the DID resolution algorithm from the [did:web Method Specification](https://w3c-ccg.github.io/did-method-web/#read-resolve) (<https://w3c-ccg.github.io/did-method-web/#read-resolve>). Third-party libraries may have features which ‘resolves’ the DID – this means turning the `kid` into the URL for the DID document and then downloading the DID. However, you must not use a third-party library’s DID resolution. This could make your application vulnerable to trusting an invalid identity.

You should only trust the DID documents located at:

- integration – <https://identity.integration.account.gov.uk/.well-known/did.json> (<https://identity.integration.account.gov.uk/.well-known/did.json>)
- production – <https://identity.account.gov.uk/.well-known/did.json> (<https://identity.account.gov.uk/.well-known/did.json>)

To retrieve the DID document, you should make HTTP request to the appropriate endpoint, for example:

```
GET /.well-known/did.json HTTP/1.1
Host: identity.integration.account.gov.uk
User-Agent: my-platform/version (https://my-service-url.gov.uk)
```

 **GOV.UK One Login requires the [User-Agent header \(/before-integrating/set-user-agent-header/\)](#) to be populated. If it absent or empty, your service will receive a 403 error**

GOV.UK One Login will always publish the DID documents on the URLs above and will never change the publication URLs without notifying you.

Follow the steps below to use the `kid` to determine which public key from the DID document was used to sign the JWT. This is important because GOV.UK One Login may have rotated its public keys and using the incorrect key will break your integration.

1. Split the `kid` from the JWT header into two parts: the controller ID (before the `#`) and the unique key ID (after the `#`). For example, in the `kid did:web:identity.integration.account.gov.uk#c9f8da1c87525bb41653583c2d05274e85805ab7d0abc58376c7128129daa936`, the controller ID is `did:web:identity.integration.account.gov.uk` and the unique key ID is `c9f8da1c87525bb41653583c2d05274e85805ab7d0abc58376c7128129daa936`.
2. Download the DID document from the DID endpoint you need:
  - o Integration: [\(https://identity.integration.account.gov.uk/.well-known/did.json\)](https://identity.integration.account.gov.uk/.well-known/did.json)
  - o Production: [\(https://identity.account.gov.uk/.well-known/did.json\)](https://identity.account.gov.uk/.well-known/did.json).
3. Make sure the controller ID matches the `id` in the DID document.
4. Find the object in `assertionMethods` which has an `id` field matching the `kid` from the JWT header. If there are multiple keys in the DID document, GOV.UK One Login is in the process of rotating its keys. If there's a key without a matching `id`, do not trust the identity and contact GOV.UK One Login to report an incident.
5. Use the `publicKeyJwk` object of the key you want to use to verify the signature.

## Cache the DID document

You should cache the returned DID document and re-use it instead of downloading the DID document for every signature you need to verify. The DID document will not change often and caching it reduces latency for your service.

The `Cache-Control` HTTP header field in the DID endpoint contains a suggested caching period. This caching period is how long GOV.UK One Login expects the DID document to remain valid.

For example, a header with the value `Cache-Control: max-age=3600, private...` would mean you cache the DID document for the `max-age` of 1 hour (3600 seconds = 1 hour). `private` stops any other caches or proxies from caching the DID document.

Occasionally, you may not be able to refresh the cache from GOV.UK One Login's URL, for example if there's a temporary outage. If this happens, you should continue to trust the cached version until you're able to refresh the cache.

For more details on the `Cache-Control` header, see [RFC 9111: HTTP Caching](https://www.rfc-editor.org/rfc/rfc9111#field.cache-control) (<https://www.rfc-editor.org/rfc/rfc9111#field.cache-control>).

## Understand the core identity signing key rotations

GOV.UK One Login will rotate its keys for the:

- integration environment - weekly from 29 October 2024 so you can test your integration
- production environment - every 6 months starting from 30 January 2025

GOV.UK One Login may need to rotate keys at short notice, for example if a key is compromised. New public keys will appear in the `assertionMethod` array of the DID document before any rotation.

Use the `Cache-Control` headers and [guidance on caching the DID document \(/integrate-with-integration-environment/prove-users-identity/#cache-the-did-document\)](#) to regularly poll the DID endpoint to detect new versions and make sure you're using the latest key.

Once GOV.UK One Login has removed the old public key from the DID document, it will no longer be valid. You should no longer trust verifiable credentials signed with that key.

## Validate your user's identity credential

1. You must validate the JWT signature according to the [JSON Web Signature Specification](#) (<https://datatracker.ietf.org/doc/html/rfc7515>). Check the JWT `alg` header is `ES256` and then use the value of the JWT `alg` header parameter to validate the JWT.
2. Check the `iss` claim is <https://identity.integration.account.gov.uk/>.
3. Check the `aud` claim matches your client ID you received when you [registered your service to use GOV.UK One Login \(/before-integrating/register-and-manage-your-service/\)](#).
4. Check the `sub` claim matches the `sub` claim you received in [the id\\_token from your token request](#).
5. Check the current time is before the time in the `exp` claim.

## Check your user's level of authentication protection matches the requested level

You must look for the `vot` (Vector of Trust) claim in the ID token and make sure the level of protection matches or exceeds the level a user needs to access your service. The `vot` claim will only contain the credential trust level, not the level of confidence, even if you make an identity request. Additionally, if you ask for medium confidence ( `P2` ) you must also request a protection level of `C1.Cm`. This means logging in with two-factor authenticaion. If you do not do this, you'll receive the error: `invalid_request - Request vtr not valid`.

## Process your user's identity credential

The identity credential contains the following claims as properties of `credentialSubject`.

Property	Description
----------	-------------

<b>name</b>	A list showing the names proven by GOV.UK One Login. This list reflects name changes by using the <code>validFrom</code> and <code>validUntil</code> metadata properties. If <code>validUntil</code> is <code>null</code> or not present, that name is your user's current name. If <code>validFrom</code> is <code>null</code> or not present, your user may have used that name from birth.
	Each name is presented as an array in the <code>nameParts</code> property. Each part of the name is either a <code>GivenName</code> or a <code>FamilyName</code> , identified in its <code>type</code> property. The <code>value</code> property could be any text string. GOV.UK One Login cannot specify a maximum length or restrictions on what characters may appear.
	<code>GivenName</code> or <code>FamilyName</code> can appear in any order within the list. The order of names may depend on either your user's preferences or the order they appear on documents used to prove your user's identity.
<b>birthDate</b>	A list of <a href="https://schema.org/Date">ISO 8601 date (https://schema.org/Date)</a> strings. There may be multiple dates of birth, for example, if there's evidence an incorrect date of birth was previously recorded for your user. The date of birth GOV.UK One Login has highest confidence in will be the first item in the list.

## Understand your user's address claim

The <https://vocab.account.gov.uk/v1/address> claim contains all addresses your user has entered, including previous addresses. GOV.UK One Login checks the address format and performs some checks, depending on the address's location and user's journey.

GOV.UK One Login collects at least 3 months' worth of addresses.

GOV.UK One Login supports the following characters in the address claim:

- digits: `0-9`
- letters: `A-Z` and `a-z`
- special characters: `'` (apostrophe), `.` (period), `,` (comma), `\` (backslash), `/` (forward slash), `*` (asterisk), and `-` (hyphen)
- space:

Each JSON object in the list may contain any of the following properties:

Property	Definition
----------	------------

<b>validFrom</b>	<p><a href="https://schema.org&gt;Date">ISO 8601 date (https://schema.org/Date)</a> strings representing the date your user moved into the address.</p> <p>GOV.UK One Login only collects the year from the user. The month and day will always default to <code>01</code>. For example, if the user moved in 2024, the <code>validFrom</code> date would be <code>2024-01-01</code>.</p>
<b>validUntil</b>	<p><a href="https://schema.org&gt;Date">ISO 8601 date (https://schema.org/Date)</a> strings representing the date your user moved from the address. This property is not included for your user's current address.</p> <p>GOV.UK One Login only collects the year from the user. The month and day will always default to <code>01</code>. For example, if the user moved in 2024, the <code>validUntil</code> date would be <code>2024-01-01</code>.</p> <p>If a user tells us an address is their current address, then <code>validUntil</code> will not be returned.</p>
<b>uprn</b>	<p>GOV.UK One Login will provide a <a href="https://www.gov.uk/government/publications/open-standards-for-government/identifying-property-and-street-information">Unique Property Reference Number (UPRN)</a> (<a href="https://www.gov.uk/government/publications/open-standards-for-government/identifying-property-and-street-information">https://www.gov.uk/government/publications/open-standards-for-government/identifying-property-and-street-information</a>) for UK addresses only.</p> <p>If a user has edited their address, the UPRN field will automatically clear.</p>
<b>organisationName</b>	<p>Maps to <code>ORGANISATION_NAME</code> in the <a href="https://www.royalmail.com/find-a-postcode">Postcode Address File</a> (<a href="https://www.royalmail.com/find-a-postcode">https://www.royalmail.com/find-a-postcode</a>) and <a href="https://apidocs.os.uk/docs/os-places-dpa-output">Ordnance Survey Places API</a> (<a href="https://apidocs.os.uk/docs/os-places-dpa-output">https://apidocs.os.uk/docs/os-places-dpa-output</a>).</p>
<b>departmentName</b>	<p>Maps to <code>DEPARTMENT_NAME</code> in the <a href="https://www.royalmail.com/find-a-postcode">Postcode Address File</a> (<a href="https://www.royalmail.com/find-a-postcode">https://www.royalmail.com/find-a-postcode</a>) and <a href="https://apidocs.os.uk/docs/os-places-dpa-output">Ordnance Survey Places API</a> (<a href="https://apidocs.os.uk/docs/os-places-dpa-output">https://apidocs.os.uk/docs/os-places-dpa-output</a>).</p>
<b>subBuildingName</b>	<p>Maps to <code>SUB_BUILDING_NAME</code> in the <a href="https://www.royalmail.com/find-a-postcode">Postcode Address File</a> (<a href="https://www.royalmail.com/find-a-postcode">https://www.royalmail.com/find-a-postcode</a>) and <a href="https://apidocs.os.uk/docs/os-places-dpa-output">Ordnance Survey Places API</a> (<a href="https://apidocs.os.uk/docs/os-places-dpa-output">https://apidocs.os.uk/docs/os-places-dpa-output</a>).</p> <p><code>subBuildingName</code> may accompany either <code>buildingName</code> or <code>buildingNumber</code>.</p>
<b>buildingNumber</b>	<p>Maps to <code>BUILDING_NUMBER</code> in the <a href="https://www.royalmail.com/find-a-postcode">Postcode Address File</a> (<a href="https://www.royalmail.com/find-a-postcode">https://www.royalmail.com/find-a-postcode</a>) and <a href="https://apidocs.os.uk/docs/os-places-dpa-output">Ordnance Survey Places API</a> (<a href="https://apidocs.os.uk/docs/os-places-dpa-output">https://apidocs.os.uk/docs/os-places-dpa-output</a>).</p>

buildingName	Maps to <b>BUILDING_NAME</b> in the <a href="#">Postcode Address File</a> ( <a href="https://www.royalmail.com/find-a-postcode">https://www.royalmail.com/find-a-postcode</a> ) and <a href="#">Ordnance Survey Places API</a> ( <a href="https://apidocs.os.uk/docs/os-places-dpa-output">https://apidocs.os.uk/docs/os-places-dpa-output</a> ).
dependentStreetName	Maps to <b>DEPENDENT_THOROUGHFARE_NAME</b> in the <a href="#">Postcode Address File</a> ( <a href="https://www.royalmail.com/find-a-postcode">https://www.royalmail.com/find-a-postcode</a> ) and <a href="#">Ordnance Survey Places API</a> ( <a href="https://apidocs.os.uk/docs/os-places-dpa-output">https://apidocs.os.uk/docs/os-places-dpa-output</a> ).
streetName	Maps to <b>THOROUGHFARE_NAME</b> in the <a href="#">Postcode Address File</a> ( <a href="https://www.royalmail.com/find-a-postcode">https://www.royalmail.com/find-a-postcode</a> ) and <a href="#">Ordnance Survey Places API</a> ( <a href="https://apidocs.os.uk/docs/os-places-dpa-output">https://apidocs.os.uk/docs/os-places-dpa-output</a> ).
doubleDependentAddressLocality	Maps to <b>DOUBLE_DEPENDENT_LOCALITY</b> in the <a href="#">Postcode Address File</a> ( <a href="https://www.royalmail.com/find-a-postcode">https://www.royalmail.com/find-a-postcode</a> ) and <a href="#">Ordnance Survey Places API</a> ( <a href="https://apidocs.os.uk/docs/os-places-dpa-output">https://apidocs.os.uk/docs/os-places-dpa-output</a> ).
dependentAddressLocality	Maps to <b>DEPENDENT_LOCALITY</b> in the <a href="#">Postcode Address File</a> ( <a href="https://www.royalmail.com/find-a-postcode">https://www.royalmail.com/find-a-postcode</a> ) and <a href="#">Ordnance Survey Places API</a> ( <a href="https://apidocs.os.uk/docs/os-places-dpa-output">https://apidocs.os.uk/docs/os-places-dpa-output</a> ).
addressLocality	Maps to <b>POST_TOWN</b> in the <a href="#">Postcode Address File</a> ( <a href="https://www.royalmail.com/find-a-postcode">https://www.royalmail.com/find-a-postcode</a> ) and <a href="#">Ordnance Survey Places API</a> ( <a href="https://apidocs.os.uk/docs/os-places-dpa-output">https://apidocs.os.uk/docs/os-places-dpa-output</a> ).
addressRegion	Maps to <a href="#">schema:addressRegion</a> ( <a href="https://schema.org/addressRegion">https://schema.org/addressRegion</a> ). Only returned for international addresses and will contain the region, provided as text. For example, California or another appropriate first-level Administrative division.
postalCode	Maps to <b>POST_CODE</b> in the <a href="#">Postcode Address File</a> ( <a href="https://www.royalmail.com/find-a-postcode">https://www.royalmail.com/find-a-postcode</a> ) and <a href="#">Ordnance Survey Places API</a> ( <a href="https://apidocs.os.uk/docs/os-places-dpa-output">https://apidocs.os.uk/docs/os-places-dpa-output</a> ).
addressCountry	Two-letter <a href="#">ISO 3166-1 alpha-2 country code</a> ( <a href="https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2">https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2</a> ).

 **The attributes might be returned in any order.**

Do not assume address properties always map to the same line of an address. For example, `addressLocality` may map to a different line of an address, depending on whether other properties are present (in this case, `dependentAddressLocality` and `doubleDependentAddressLocality`).

A sample UK ( `GB` ) address returned would look similar to this:

```
{  
    uprn: "100021051133",  
    organisationName: "Acme Corporation",  
    departmentName: "Sales Department",  
    subBuildingName: "Unit 3B",  
    buildingNumber: "42",  
    buildingName: "Riverside House",  
    dependentStreetName: "Industrial Estate",  
    streetName: "River Lane",  
    doubleDependentAddressLocality: "Riverside",  
    dependentAddressLocality: "Newtown",  
    addressLocality: "Birmingham",  
    postalCode: "B12 8QT",  
    addressCountry: "GB"  
    "validFrom": "2000-01-01",  
}
```

A sample international address returned would look similar to this:

```
{  
    "subBuildingName": "1",  
    "buildingNumber": "27",  
    "buildingName": "The Big Building",  
    "streetName": "Long Street",  
    "addressLocality": "Los Angeles",  
    "addressRegion": "California"  
    "postalCode": "90012",  
    "addressCountry": "US",  
    "validFrom": "2000-01-01",  
}
```

## Understand your user's passport claim

The <https://vocab.account.gov.uk/v1/passport> claim contains the details of your user's passport, if they submitted one when proving their identity.

`documentNumber` The passport number.

`icaoIssuerCode` An identifier for the state or organisation that issued the passport. This is defined by the International Civil Aviation Organization (ICAO) standard [9303 Machine Readable Travel Documents](#) ([https://www.icao.int/publications/Documents/9303\\_p3\\_cons\\_en.pdf](https://www.icao.int/publications/Documents/9303_p3_cons_en.pdf)). The identifier is up to 3 characters.

`expiryDate` The expiration date as an [ISO 8601 date](#) (<https://schema.org/Date>) string.

## Understand your user's driving licence claim

The `https://vocab.account.gov.uk/v1/drivingPermit` claim contains the details of your user's driving licence, if they submitted one when proving their identity.

### Property      Definition

`expiryDate` The expiry date of the driving licence as an [ISO 8601 date](#) (<https://schema.org/Date>) string.

`issueNumber` The last 2 characters of the driving licence number – these show how many times the user has received a new driving licence. You'll only receive this property for licences issued by the Driver and Vehicle Licensing Agency (DVLA).

`issuedBy` The organisation that issued the driving licence.

`personalNumber` The driver number of the driving licence. This is a string unique to the user.

## Understand your user's return code claim

 We recommend requesting the return code claim to make your error handling more clear.

To use the `returnCode` claim, you'll need to:

1. Enable the `resultCode` claim when you register your service.
2. Include `https://vocab.account.gov.uk/v1/resultCode` when you make a request for authentication and identity (<https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/authenticate-your-user/#make-a-request-for-authentication-and-identity>).

The `https://vocab.account.gov.uk/v1/resultCode` claim gives information about any issues with the evidence your user provided to prove their identity. For example, if GOV.UK One Login was not able to prove your user's identity.

When you use this claim and there's an issue with the evidence your user provided to prove their identity:

1. You'll receive an authorisation code in the `redirect_uri` instead of an `access_denied` error.
2. Use this authorisation code to get an ID token and an access token.
3. When you make a request to the `/userinfo` endpoint using the access token, the response may contain only authentication data, and an array of one or more `resultCode` values, which will each be a letter.
4. For security reasons, you'll need to contact GOV.UK One Login on [govuk-one-login@digital.cabinet-office.gov.uk](mailto:govuk-one-login@digital.cabinet-office.gov.uk) for more detailed information on what issue each `resultCode` value stands for.

Currently, there are 9 `resultCode` values which GOV.UK One Login could return if there's an issue with the evidence your user provided to prove their identity. You may receive a return code even if a user's identity verification is successful, for example, if a user is a politically exposed person. Contact GOV.UK One Login on [govuk-one-login@digital.cabinet-office.gov.uk](mailto:govuk-one-login@digital.cabinet-office.gov.uk) for more detailed information on what each return code means.

## Property Definition

---

<code>code</code>	An array of single letter codes for <code>resultCode</code> values.
-------------------	---

You can use these codes to identify the reason(s) for any issues that occurred during the identity proving journey. For security reasons, you'll need to contact GOV.UK One Login on [govuk-one-login@digital.cabinet-office.gov.uk](mailto:govuk-one-login@digital.cabinet-office.gov.uk) for more detailed information on what each return code means.

---

If you want to add this feature to an existing integration, contact GOV.UK One Login on [govuk-one-login@digital.cabinet-office.gov.uk](mailto:govuk-one-login@digital.cabinet-office.gov.uk) to update your client registration. You must also update your code to make sure your integration is able to use the new behaviour.

HTTP/1.1 200 OK

Content-Type: application/json

```
{  
  "sub": "urn:fdc:gov.uk:2022:56P4CMsGh_02Y01Wpd8PA0I-2sVlB2nsNU7mcLZYhYw=",  
  "email": "test@example.com",  
  "email_verified": true,  
  "phone_number": "+441406946277",  
  "phone_number_verified": true,  
  "https://vocab.account.gov.uk/v1/returnCode": [  
    {  
      "code": "B"  
    },  
    {  
      "code": "C"  
    }  
  ]  
}
```

Continue to [managing your users' sessions \(https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/managing-your-users-sessions/#managing-your-users-39-sessions\)](https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/managing-your-users-sessions/#managing-your-users-39-sessions).

This page was last reviewed on 12 February 2025.

[View source](#) [Report problem](#) [GitHub Repo](#)



[Accessibility](#)

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Integrate with GOV.UK One Login's integration environment

Before you can use GOV.UK One Login, you need to build a proof of concept client and explore the end-to-end journey in our integration environment. This will help you understand how to integrate with GOV.UK One Login, and where it will fit within your service.

If your service requires identity proving, you must authenticate your users first.

1. [Authenticate your user \(/integrate-with-integration-environment/authenticate-your-user/\)](#).
2. [Prove your user's identity \(/integrate-with-integration-environment/prove-users-identity/\)](#).
3. [Manage your user's session \(/integrate-with-integration-environment/managing-your-users-sessions\)](#).

To get started, you'll need to [authenticate your users \(/integrate-with-integration-environment/authenticate-your-user/\)](#).

This page was last reviewed on 2 May 2024.

[View source](#) [Report problem](#) [GitHub Repo](#)



[Accessibility](#)



[Table of contents](#)

# Quick start

Using this page is optional but can be helpful to see how a typical integration with GOV.UK One Login works.

You'll create an example service using either a local copy of the [GOV.UK One Login simulator \(/test-your-integration/gov-uk-one-login-simulator/\)](#) or the GOV.UK One Login integration environment.

You'll be able to test authentication-only or authentication and identity journeys.

You have 3 different options to create an example service, depending on your needs and how much code you want to view.

Method to run the example service	Approximate time	Result
With the GOV.UK One Login simulator using Docker Compose.	3 minutes	You'll see the simulated response from GOV.UK One Login without viewing additional code.
With the GOV.UK One Login simulator using source code.	10 minutes	You'll see the simulated response from GOV.UK One Login and view additional code.
Using the GOV.UK One Login integration environment.	20 minutes	You can use test user data to interact with the integration environment.

There's further guidance on using the [GOV.UK One Login simulator \(/test-your-integration/gov-uk-one-login-simulator/\)](#) to test your service before you use the GOV.UK One Login integration environment.

## Prerequisites

1. If you do not already have it, [install git](https://github.com/git-guides/install-git) (<https://github.com/git-guides/install-git>).
2. If you do not already have it, [install Docker Desktop](https://docs.docker.com/get-started/get-docker/) (<https://docs.docker.com/get-started/get-docker/>) (you'll use this to run the simulator).
3. [Check you are on v4.34 or higher for Docker Desktop](https://www.docker.com/blog/how-to-check-docker-version/) (<https://www.docker.com/blog/how-to-check-docker-version/>).
4. [Enable Docker Host networking](https://docs.docker.com/engine/network/drivers/host/#docker-desktop) (<https://docs.docker.com/engine/network/drivers/host/#docker-desktop>).
5. [Install nvm](https://github.com/nvm-sh/nvm) (<https://github.com/nvm-sh/nvm>).

## Run the example service with the GOV.UK One Login simulator using Docker Compose

1. On the command line, run `git clone https://github.com/govuk-one-login/onboarding-examples && cd onboarding-examples/clients/nodejs`. This will get the example Typescript code and set your working directory.
2. On the command line, run `docker compose up`.
3. Open `http://localhost:8080`.
4. Select **Make a request for authentication**.
5. If you want to run an identity journey, select **Make a request for authentication and identity**.
6. Select the **Sign out** link in the top header.

## Run the example service with the GOV.UK One Login simulator using source code

1. On the command line, run `git clone https://github.com/govuk-one-login/onboarding-examples && cd onboarding-examples/clients/nodejs`. This will get the example Typescript code and set your working directory.
2. Run `nvm install 22.11.0 && nvm use 22.11.0`. This makes sure you're using the correct version of Node.js.
3. Run `npm run simulator:start` to start the simulator in a Docker container.
4. Check the simulator is working by running `npm run simulator:config`. You should see the simulator configuration appear.
5. Run `npm ci && npm run dev:sim` to build and run the example.
6. View the example service by going to `http://localhost:8080` in your browser.
7. Select **Make a request for authentication**. You may want to use your browser's developer tools to view the web traffic, including the request to the `/authorize` endpoint and its response.
8. You should see the response from the `/userinfo` and `/token` endpoints: ID and access tokens and user attributes.

9. If you want to run an identity journey, select **Verify** again and you should see a successful identity response including the `coreIdentityJWT`, `returnCode` (empty), `address` claims.
10. Select **Sign out** in the top header.
11. You'll see a page which says **Logged out**.

## Run the example service using the GOV.UK One Login integration environment

Before you start, make sure you have a:

- recognised government email address (<https://admin.sign-in.service.gov.uk/register/enter-email-address>)
- UK mobile phone with a number starting `07` or `+44`

## Run an authentication journey using the GOV.UK One Login integration environment

### Configure the integration environment

1. On the command line, run `git clone https://github.com/govuk-one-login/onboarding-examples && cd onboarding-examples/clients/nodejs`. This will get the example Typescript code and set your working directory.
2. Run `nvm install 22.11.0 && nvm use 22.11.0`. This makes sure you're using the correct version of Node.js.
3. Run `npm run generatekeys`. This generates a key pair if one does not exist yet.
4. Launch the GOV.UK One Login admin tool (<https://admin.sign-in.service.gov.uk/register/enter-email-address>).
5. Follow on-screen instructions to register and manage your service (<https://docs.sign-in.service.gov.uk/before-integrating/register-and-manage-your-service/>) in the integration environment.
6. Configure your service name or names as `onboarding-example - {DEPARTMENT} - {SERVICE_TEAM_NAME}`
7. Find your `Client ID` value and make a record of it. You'll need this later when configuring the example application.
8. Configure your service including (at a minimum):
  - a redirect URI: `http://localhost:8080/oidc/authorization-code/callback`
  - a public key (copy the static public key you created earlier from the `./public_key.pem` file, excluding the headers)
  - scopes: `openid`, `email`, `phone`
  - a post logout redirect URI: `http://localhost:8080/signed-out`
  - there's further guidance on registering and managing your service ([/before-integrating/register-and-manage-your-service/#register-and-manage-your-service](#)) if you

want to include additional fields

## Configure the example application

1. Create a `.env.integration` configuration file by copying the `.env.integration.example` file to `.env.integration`.
2. Edit `.env.integration` in your preferred source editor and update:
  - the `{CLIENT_ID}` placeholder to contain the Client ID from the GOV.UK One Login admin tool
  - the `{PRIVATE_KEY}` placeholder with the contents of the `./private_key.pem` file you created earlier (excluding the headers and removing the line breaks)

## Start the example application and follow the journey

1. Run `npm ci && npm run dev:int` – this installs the dependencies and runs the application.
2. View the example service by going to `http://localhost:8080` in your browser.
3. Select **Make a request for authentication**. You may want to use your browser's developer tools to view the web traffic, including the request to the `/authorize` endpoint and its response.
4. Follow the on-screen instructions to create a GOV.UK One Login.
5. You should see the response from the `/userinfo` and `/token` endpoints: ID and access tokens and user attributes.

If you want to run an authentication-only journey, you can stop here.

## Run an authentication and identity journey using the GOV.UK One Login integration environment

If you want to run an authentication and identity journey, you should do the following additional steps as well as the steps above.

1. Update your client configuration in the integration environment using the [GOV.UK One Login admin tool](https://admin.sign-in.service.gov.uk/register/enter-email-address) (<https://admin.sign-in.service.gov.uk/register/enter-email-address>):
  - set **Prove user's identities** to Yes
  - set the claims to `coreIdentityJWT`, `resultCode` and `address`
2. [Follow the guidance to test a successful identity proving journey](#) ([/test-your-integration/using-integration-for-testing/#test-a-successful-identity-proving-journey](#)), starting at step 3.
3. You need to request fictional users and their knowledge-based verification (KBV) answers to help you test your journeys. [Contact GOV.UK One Login](#) to access this test user data.
4. Using this test user data, you should see a successful identity response including the `coreIdentityJWT`, `resultCode` (empty), `address` claims. If you do not, [get in touch](#)

If you have any issues:

- [get in touch on the govuk-one-login-tech-support Slack channel](https://ukgovernmentdigital.slack.com/archives/C02K303R44R)  
[\(https://ukgovernmentdigital.slack.com/archives/C02K303R44R\)](https://ukgovernmentdigital.slack.com/archives/C02K303R44R)
- [contact GOV.UK One Login on email](#)

This page was last reviewed on 26 February 2025.

[View source](#) [Report problem](#) [GitHub Repo](#)



[Accessibility](#)

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Support

[Use the #govuk-one-login channel](#)

(<https://ukgovernmentdigital.slack.com/archives/C02AQUJ6WTC>) to contact the GOV.UK One Login technical team.

This page was last reviewed on 2 December 2022.

[View source](#) [Report problem](#) [GitHub Repo](#)



[Accessibility](#)

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Test your service with the GOV.UK One Login simulator

If you're a service developer you can use the GOV.UK One Login simulator to:

- build your service with your choice of development environment and frameworks
- test your service locally with a range of data, return codes and error scenarios

If you're a quality assurance tester you can use the GOV.UK One Login simulator to perform end to end testing of your service with your own pre-configured data.

With the GOV.UK One Login simulator you can:

- test and verify specific user information, such as names and email addresses
- request specific error scenarios and write code to handle these
- test responses for identity verification without going through the identity proving process

You can run the simulator locally. It is distributed as a Docker image from the [GitHub container registry](#) (<https://github.com/govuk-one-login/simulator/pkgs/container/simulator/versions>).

The GOV.UK One Login team runs daily acceptance tests against the live system, so you'll always be using the most up-to-date API schemas.

 **The GOV.UK One Login simulator is not the real GOV.UK One Login. Before you go live you must [test your application using the integration environment](#) ([/test-your-integration/using-integration-for-testing/](#)).**

## API endpoints

The simulator exposes the following API endpoints:

Endpoint	Description
----------	-------------

---

/	A simulator endpoint that confirms it is running by displaying Express + TypeScript Server .
<code>/.well-known/openid-configuration</code>	An OpenID configuration endpoint.
<code>/.well-known/jwks.json</code>	A JSON Web Keys (JWKS) endpoint to publish the public keys that sign the ID token.
<code>/.well-known/did.json</code>	A decentralised identifier (DID) endpoint to publish the public keys that sign the core identity.
<code>/authorize</code>	An OpenID Connect (OIDC) endpoint to An OpenID Connect (OIDC) endpoint to <a href="https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/authenticate-your-user/#make-a-request-to-the-authorize-endpoint">authenticate the user (https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/authenticate-your-user/#make-a-request-to-the-authorize-endpoint)</a> .
<code>/config</code>	A simulator configuration endpoint for modifying and requesting the current configuration using <code>POST</code> and <code>GET</code> .
<code>/logout</code>	An OIDC endpoint to <a href="https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/managing-your-users-sessions/#log-your-user-out-of-gov-uk-one-login">log the user out (https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/managing-your-users-sessions/#log-your-user-out-of-gov-uk-one-login)</a> .
<code>/token</code>	An OIDC endpoint to <a href="https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/authenticate-your-user/#make-a-token-request">exchange the authorisation code for tokens (https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/authenticate-your-user/#make-a-token-request)</a> .
<code>/trustmark</code>	A OIDC trustmark document listing vectors of trust implemented by GOV.UK One Login.
<code>/userinfo</code>	An OIDC endpoint to <a href="https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/authenticate-your-user/#retrieve-user-information">retrieve user information (https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/authenticate-your-user/#retrieve-user-information)</a> .

---

## Run the GOV.UK One Login simulator in Docker without configuration

If you do not already have it, [install Docker Desktop \(https://www.docker.com/products/docker-desktop/\)](https://www.docker.com/products/docker-desktop/) (version 4.34.0 or higher).

# Run the GOV.UK One Login simulator locally with Docker Desktop

1. In Docker Desktop, select the **Settings** symbol (cog) in the top right corner.
2. In Docker Desktop, select **Resources**, then **Network** from the left hand menu.
3. In Docker Desktop, select **Enable host networking**, then select **Apply & restart**.
4. On the command line, run `docker run --rm --detach --publish 3000:3000 --name simulator ghcr.io/govuk-one-login/simulator:latest`.

# Run the GOV.UK One Login simulator from source code without configuration

1. If you do not already have it, [install git \(<https://github.com/git-guides/install-git>\)](https://github.com/git-guides/install-git).
2. If you do not already have it, [install nvm \(<https://github.com/nvm-sh/nvm>\)](https://github.com/nvm-sh/nvm).

## Run the GOV.UK One Login simulator locally

1. Run `git clone https://github.com/govuk-one-login/simulator && cd simulator`. This will get the simulator Typescript code and set your working directory.
2. Run `nvm install 22.11.0 && nvm use 22.11.0`. This makes sure you're using the correct version of Node.js.
3. Run `npm install && npm run build` to build the simulator.
4. Run `npm run start` to run the simulator.
5. Check the simulator is working by running `curl http://localhost:3000`. You should see the simulator configuration appear.

You'll need to adjust your configuration to use the simulator as a replacement for the GOV.UK One Login OpenID provider, instead of `oidc.account.gov.uk` or `oidc.integration.account.gov.uk`.

## Change the GOV.UK One Login simulator's default port

The GOV.UK One Login simulator runs on `http://localhost:3000` by default.

You can run it on another port if needed. For example, to switch it to `localhost:3333` run:

```
docker run -e SIMULATOR_URL='http://localhost:3333' -e PORT=3333 --rm -ti -
```

If you're not using Docker you can run:

```
PORT=3333 SIMULATOR_URL=http://localhost:3333 npm run start
```

# View the default configuration

To check the default configuration of the simulator run:

- ▶ Show command

This table shows the default configuration values:

Field	Default value
clientId	HGI0gho9HIRhgoepdIOPFdIUWgewi0jw
publicKey	-----BEGIN PUBLIC KEY----- MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAmXXR3EsRvUMVhEJMtQ1we xJjfQ00Q0MQ7ARfShN53Bn0QEPFnS/I8ntBddkKdE3q+vMTI72w6Fv3SsMM+ciR2L IHdEQfKgsLt6PGNcV1kG6GG/3nSW3psW8w65Q3fmy81P1748qezDrVfaGrF4PDXAL zX1ph+nz8mpKmck6aY6LEUJ4B+TIfYz1KmmwFe3ri0spSW+J5wE9mmT3Vkr2ySuHR YHqlx1F9dfX7l0TsbgJFzN6T001ZQDhY0iLwzdGwhSx06R6N/ZINYHCKFPaQD+td Ksrw7QDIYnx0IiXFnkGnizl3UtqSmXAaceTvPM2Pz84x2JiwHrp2Sm16RYLCQIDAQ AB-----END PUBLIC KEY-----
scopes	["openid", "email", "phone"]
redirectUrls	["http://localhost:8080/oidc/authorization-code/callback"]
claims	["https://vocab.account.gov.uk/v1/coreIdentityJWT", "https://vocab.account.gov.uk/v1/address", "https://vocab.account.gov.uk/v1/returnCode"]
identityVerificationSupported	true
idTokenSigningAlgorithm	ES256

clientLoCs	["P0", "P2"]
sub	urn:fdc:gov.uk:2022:56P4CMsGh_02Y0lWpd8PAOI-2sVlB2nsNU7mcLZYhYw=
email	test@example.com
emailVerified	true
phoneNumber	07123456789
phoneNumberVerified	true
maxLocationAchieved	P2
coreIdentityVerifiableCredentials	{"type": ["VerifiableCredential", "IdentityCheckCredential"], "credentialSubject": {"name": [{"nameParts": [{"value": "GEOFFREY", "type": "GivenName"}, {"value": "HEARNSHAW", "type": "FamilyName"}]}], "birthDate": [{"value": "1955-04-19"}]}
passportDetails	null
drivingPermitDetails	null
postalAddressDetails	{"postalAddressDetails": [{"addressCountry": "GB", "buildingName": "", "streetName": "FRAMPTON ROAD", "postalCode": "GL1 5QB", "buildingNumber": "26", "addressLocality": "GLOUCESTER", "validFrom": "2000-01-01", "uprn": "100120472196", "subBuildingName": ""}]}
returnCodes	null
simulationUrl	http://localhost:3000

```
postLog      ["http://localhost:8080/signed-out"]
outRedir
ectUrls
```

---

The private key for the default public key is:

- ▶ Show private key

The GOV.UK One Login simulator is also set up with a default private/public key pair for client assertion. The private key for the default key pair is:

- ▶ Show private key

## Change the default configuration

You can change the client configuration or use a different key to sign your client assertion. You do this by setting environment variables when running the simulator or send a [POST](#) request to the `/config` endpoint in the format:

```
```
{
  "clientConfiguration": {
    "redirectUrls": ["http://localhost:8080/callback"],
    "idTokenSigningAlgorithm": "RS256",
    "publicKey": "TEST_PUBLIC_KEY"
  }
}
````
```

## Configure the GOV.UK One Login simulator

### Set up client, response and error configuration

There are 3 ways you can set up the client, response and error configuration for the GOV.UK One Login simulator:

1. Use environment variables – these work best if you have a static configuration which should not change frequently.
2. Make a `POST` request to the `/config` endpoint to update the configuration – this works best for a configuration which you are likely to update frequently.
  - a `POST` request to the `/config` endpoint will overwrite any fields set as environment variables while the Docker container is running.
3. Set the environment variable `INTERACTIVE_MODE` to `true`. This is best if you want to [return multiple response configurations](#) ([/test-your-integration/gov-uk-one-login-simulator/#returning-multiple-response-configurations](#)).

There are [examples of how to send the simulator requests on GitHub](#) (<https://github.com/govuk-one-login/onboarding-examples/tree/main/data/simulator-configuration>).

Parameters provided as environment variables which are parsed as an array should be set as a comma-separated string, for example `SCOPES=openid,email`.

If you input invalid configuration fields, the simulator might:

- not use them
- return an error
- return unexpected results

## Reset GOV.UK One Login simulator back to its default settings

To reset the GOV.UK One Login simulator configuration back to its default settings, you need to stop the container in Docker and restart it.

If you're not using Docker you can stop the GOV.UK One Login simulator by running `ctrl+C` on the command line and restarting the GOV.UK One Login simulator.

## Configure the client

When updating the client configuration using the `/config` endpoint, you must use the following JSON structure in the request body:

```
{  
  "clientConfiguration": {  
    "clientId": "ClientId",  
    "scopes": ["openid", "phone", "email"],  
    ...other fields  
  },  
}
```

This table describes the different fields for the client configuration:

| Environment variable             | Config request field          | Description  | Valid values  |
|----------------------------------|-------------------------------|--|---|
| CLAIMS                           | claims                        | The claims you configured the client to request.   | <ul style="list-style-type: none"> <li>• <a href="https://vocab.account.gov.uk/v1/passport">https://vocab.account.gov.uk/v1/passport</a></li> <li>• <a href="https://vocab.account.gov.uk/v1/address">https://vocab.account.gov.uk/v1/address</a></li> <li>• <a href="https://vocab.account.gov.uk/v1/drivingPermit">https://vocab.account.gov.uk/v1/drivingPermit</a></li> <li>• <a href="https://vocab.account.gov.uk/v1/coreId/entityJWT">https://vocab.account.gov.uk/v1/coreId/entityJWT</a></li> <li>• <a href="https://vocab.account.gov.uk/v1/returnCode">https://vocab.account.gov.uk/v1/returnCode</a></li> </ul> |
| CLIENT_ID                        | clientId                      | The public identifier for a client.  | Any string  |
| CLIENT_LOCS                      | clientLocs                    | The levels of confidence values which the client can request.                                  | P0 , P1 , P2  |
| IDENTITY_VERIFICATION_SUPPORTERD | identityVerificationSupported | Whether or not the client has identity verification enabled.                                   | Boolean   |
| ID_TOKEN_SIGNING_ALGORITHM       | idTokenSigningAlgorithm       | The algorithm which you should sign the ID token with.   | ES256 or RS256  |
| PUBLIC_KEY                       | publicKey                     | The public key the simulator will use to validate the <code>client_assertion</code> signature. | PEM-encoded public key  |

|                      |                           |  |   |
|----------------------|---------------------------|--|---|
| <b>REDIRECT_URLS</b> | <code>redirectUrls</code> | The redirect URLs, which your users will be redirected to. | Any valid URLs  |
| <b>SCOPES</b>        | <code>scopes</code>       | The scopes you've configured the client to request.        | <ul style="list-style-type: none"> <li>• <code>openid</code></li> <li>• <code>email</code></li> <li>• <code>phone</code></li> </ul> |

## Configure the response

When updating the response configuration using the `/config` endpoint, you must use the following JSON structure in the request body:

```
{
  "responseConfiguration": {
    "sub": "someSubjectIdentifier",
    "email": "anExampleEmail@example.com" ,
    ...other fields
  },
}
```

This table describes the different fields for the response configuration:

| Environment variable | Config request field                           | Description   | Valid values |
|----------------------|--|---|--------------|
| N/A                  | <code>coreIdentityVerifiableCredentials</code> | A core identity verifiable credential.                              | JSON object  |
| N/A                  | <code>drivingPermitDetails</code>              | A set of driving licence details the simulator returns to the user. | JSON array   |
| EMAIL                | <code>email</code>                             | The returned email address.   | Any string   |
| EMAIL_VERIFIED       | <code>emailVerified</code>                     | Whether or not the email address has been verified.                 | Boolean      |
| N/A                  | <code>maxLocAchieved</code>                    | The maximum level of confidence the user achieved.                  | Any string   |

|                       |                                   |  |   |
|-----------------------|-----------------------------------|--|---|
| N/A                   | <code>passportDetails</code>      | A set of passport details the simulator returns to the user.   | JSON array  |
| PHONE_NUMBER          | <code>phoneNumber</code>          | The returned phone number.   | Any string  |
| PHONE_NUMBER_VERIFIED | <code>phoneNumberVerified</code>  | Whether or not the phone number has been verified.   | Boolean   |
| N/A                   | <code>postalAddressDetails</code> | A set of address details the simulator returns to the user.  | JSON array  |
| N/A                   | <code>returnCodes</code>          | A set of codes returned if the return code claim is included in the client configuration and <code>/authorize</code> request. Otherwise an <code>ACCESS_DENIED</code> error will return when this is configured. | JSON array with the following structure<br><code>[{"code": "anyString"}]</code> |
| SUB                   | <code>sub</code>                  | The returned pairwise subject identifier.  | Any string  |

If the valid values are JSON objects or JSON arrays, no further validation is done on the provided response configuration unless outlined. You can see example data in the [GOV.UK One Login onboarding README](https://github.com/govuk-one-login/onboarding-examples/tree/main/data) (<https://github.com/govuk-one-login/onboarding-examples/tree/main/data>).

## Configure the errors

You can set up the simulator to return specific error scenarios at the `/authorize` endpoint as well as in the core identity JSON Web Token (JWT) and the ID token.

There are no defaults configured for the error configuration, so you must provide these if you want the simulator to return an error.

You can set multiple error states, which you can pass as a comma-separated string to these environment variables:

- `CORE_IDENTITY_ERRORS`
- `ID_TOKEN_ERRORS`
- `AUTHORISE_ERRORS`

Alternatively, you can set multiple error states using the `/config` endpoint with the following syntax:

```
{  
  "errorConfiguration": {  
    "coreIdentityErrors": ["INVALID_ALG_HEADER"],  
    "idTokenErrors": ["INVALID_ISS"],  
    "authoriseErrors": ["ACCESS_DENIED"]  
  }  
}
```

The simulator will ignore any invalid values for the error configuration.

#### /authorize endpoint errors configurable on the GOV.UK One Login simulator

These are errors returned by the GOV.UK One Login simulator at the point in which a user hits the `/authorize` endpoint.

| Error type | Detail |
|------------|--------|
|------------|--------|

|               |  |
|---------------|--|
| ACCESS_DENIED | See <a href="https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/authenticate-your-user/#http-302-found">Authenticate your user (https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/authenticate-your-user/#http-302-found)</a> for more information on this error message. |
|---------------|--|

#### ID token errors configurable on the GOV.UK One Login simulator

These are errors in the issued ID token returned by the GOV.UK One Login simulator.

| Error type | Detail |
|------------|--------|
|------------|--------|

|               |  |
|---------------|--|
| INCORRECT_VOT | The vector of trust ( <code>vot</code> ) returned in the token does not match the vector of trust requested ( <code>vtr</code> ) in the <code>/authorize</code> request. |
|---------------|--|

|                    |  |
|--------------------|--|
| INVALID_ALG_HEADER | The <code>alg</code> in the header does not match the algorithm returned from the <code>/jwks</code> endpoint. |
|--------------------|--|

|             |                                   |
|-------------|-----------------------------------|
| INVALID_AUD | ID token has an invalid audience. |
|-------------|-----------------------------------|

|             |                                 |
|-------------|---------------------------------|
| INVALID_ISS | ID token has an invalid issuer. |
|-------------|---------------------------------|

|                   |  |
|-------------------|--|
| INVALID_SIGNATURE | The signature of the token is invalid. |
|-------------------|--|

**NONCE\_NOT\_MATCHING** The `nonce` in the token does not match the `nonce` supplied in the `/authorize` request.

**TOKEN\_EXPIRE** The expiry date of the token is in the past.

D

**TOKEN\_NOT\_VALID\_YET** The `iat` claim of the token is in the future.

## Core identity errors configurable on the GOV.UK One Login simulator

These are errors in the issued core identity JWT returned by the GOV.UK One Login simulator.

| Core identity errors | Detail |
|----------------------|--------|
|----------------------|--------|

|                      |   |
|----------------------|---|
| <b>INCORRECT_SUB</b> | The <code>sub</code> does not match the <code>sub</code> in the <code>id_token</code> . Sub is the 'subject identifier' or the unique ID of a user. |
|----------------------|---|

|                           |  |
|---------------------------|--|
| <b>INVALID_ALG_HEADER</b> | The <code>alg</code> in the header is not <code>ES256</code> . |
|---------------------------|--|

|                    |  |
|--------------------|--|
| <b>INVALID_AUD</b> | Core identity has an invalid audience. |
|--------------------|--|

|                    |                                      |
|--------------------|--------------------------------------|
| <b>INVALID_ISS</b> | Core identity has an invalid issuer. |
|--------------------|--------------------------------------|

|                          |  |
|--------------------------|--|
| <b>INVALID_SIGNATURE</b> | The signature of the token is invalid. |
|--------------------------|--|

|                      |  |
|----------------------|--|
| <b>TOKEN_EXPIRED</b> | The expiry date of the token is in the past. |
|----------------------|--|

To remove an error configuration, you can either unset the environment variables, or you can make a `POST` request to the `/config` endpoint without the `errorConfiguration` field in the body.

If you update your configuration using the `/config` endpoint you must include the `errorConfiguration` field if you want to maintain the errors you've configured.

## Configure simulator base URL

If you want to deploy the simulator using a host name or port other than `localhost` and `3000`, you can configure the base URL where the simulator is hosted. You can also update the URL using the `/config` endpoint with the following request body field:

```
{  
  "simulatorUrl": "https://example.com:3333"  
}
```

Modifying the simulator URL will affect other endpoints and any validation that includes these endpoints. For example, the token endpoint will become ``${SIMULATOR_URL}/token``, so you need to update the expected audience of the client assertion to reflect this.

## Returning multiple response configurations

You can set the GOV.UK One Login simulator to return multiple response configurations. This can help you to test how your system handles different responses from GOV.UK One Login.

For example, you can fill in one response with passport and address data, but for another request you could swap the passport data for driving license data.

To do this, set the environment variable `INTERACTIVE_MODE` to `true`.

With `INTERACTIVE_MODE` enabled, after you make the `/authorize` request you'll see a form where you can add the expected response configuration.

You must submit the `sub` field when filling in the form. All other fields are optional.

By default, the form is pre-populated with the same [response configuration the GOV.UK One Login simulator is configured with](#) (<https://github.com/govuk-one-login/simulator/blob/main/docs/configuration.md>). You can overwrite each field with the expected values for the response configuration fields.

The form will show all possible configurable fields, even if the `/authorize` request you're submitting does not include the [scope or claims](#) (<https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/authenticate-your-user/#replace-the-placeholder-values-in-your-example>) required for the simulator to return them.

For example, your `/authorize` request might not include the `https://vocab.account.gov.uk/v1/passport` claim, but the form will still include this field. However, the GOV.UK One Login simulator will only return the scopes or claims you include in your `/authorize` request.

Any response configuration form fields that you do not submit will use the [pre-configured response fields](#) (<https://github.com/govuk-one-login/simulator/blob/main/docs/configuration.md>).

All values you submit through the form are [validated to the same level as values submitted to the '/config' endpoint \(<https://github.com/govuk-one-login/simulator/blob/main/docs/configuration.md>\)](https://github.com/govuk-one-login/simulator/blob/main/docs/configuration.md).

You can find example JSON for identity claims in the [technical documentation \(<https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/prove-users-identity/#prove-your-user-39-s-identity>\)](https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/prove-users-identity/#prove-your-user-39-s-identity) or in the [onboarding examples \(<https://github.com/govuk-one-login/onboarding-examples>\)](https://github.com/govuk-one-login/onboarding-examples).

Once you've entered the fields you'd like to test the responses for, select **Continue**. You will then be redirected to the `redirect_uri` from your `/authorize` request.

When your service exchanges the `issued access_token` at the `/userinfo` endpoint, the simulator will return the response configuration you submitted in the form.

If you submit an invalid field, you may see the following error response:

```
{  
  "error": "invalid_request",  
  "invalid_fields": [  
    {  
      "field": "a field name",  
      "msg": "an error message"  
    }  
  ]  
}
```

In this response, `field` tells you which of your submitted fields is invalid. If you see this error, check the data in the specified field. If you continue to see this error message contact the GOV.UK One Login team for support.

## Support and feedback

[Raise a GitHub Issue with the GOV.UK One Login simulator \(<https://github.com/govuk-one-login/simulator/issues>\)](https://github.com/govuk-one-login/simulator/issues) if you:

- discover a bug or an error
- struggle with any aspect of using the simulator
- would like to suggest improvements

If you have more general feedback or questions, you can get in touch with the team on our cross-government [GOV.UK One Login tech support Slack channel \(<https://ukgovernmentdigital.slack.com/archives/C02K303R44R>\)](https://ukgovernmentdigital.slack.com/archives/C02K303R44R).

This page was last reviewed on 12 June 2025.

[View source](#) [Report problem](#) [GitHub Repo](#)



[Accessibility](#)

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Using the integration environment for end-to-end testing

You can use our integration environment to test your end-to-end user journeys.

This page describes:

- what to do [before you begin testing \(/test-your-integration/using-integration-for-testing/#before-you-begin\)](#)
- how to [navigate automated testing \(/test-your-integration/using-integration-for-testing/#navigating-automated-testing\)](#)
- how to [conduct end-to-end tests against the integration environment \(/test-your-integration/using-integration-for-testing/#conducting-end-to-end-user-testing-against-the-integration-environment\)](#)
- how to [navigate internal performance testing of your service \(/test-your-integration/using-integration-for-testing/#navigating-internal-performance-testing-of-your-service\)](#)

 **You must not conduct any security testing, penetration testing, performance testing, or IT health checks of the GDS estate. You must also not use personal identifiable information (PII) – GOV.UK One Login will provide example data.**

You should focus on end-to-end testing the critical paths, for example testing a successful identity journey. There's further guidance on how to [conduct end-to-end tests against the integration environment \(/test-your-integration/using-integration-for-testing/#conducting-end-to-end-user-testing-against-the-integration-environment\)](#).

We will notify you for any changes made to the GOV.UK One Login API.

We will not notify you for changes that are internal to the GOV.UK One Login journey, for example, if the wording on a button changes.

## Before you begin

Before you can test on our integration environment, you must:

- have [registered your service to use GOV.UK One Login \(/before-integrating/register-and-manage-your-service/\)](#)
- have built an application to work with GOV.UK One Login
- have accessed the [example responses from the GOV.UK One Login API \(<https://github.com/govuk-one-login/onboarding-examples/tree/main/data>\)](#)
- have contacted GOV.UK One Login to access the fictional users and their knowledge-based verification (KBV) answers to help you test your journeys

## Navigating automated testing

GOV.UK One Login does not provide specific recommendations about automated testing. This is because we are making frequent updates to the code and user flows that may break your tests.

However, if you choose to do automated testing, you might need to generate a one-time code using a scripting language.

### Generate a one-time code using a scripting language

When conducting automated testing, the multi-factor authentication may block your automated tests. You can generate a one-time code using a scripting language to help your automated tests run as expected.

1. Go to your service start page.
2. Select **Start**.
3. Select **Create a GOV.UK One Login**.
4. Follow the instructions to create an account using the test user data. You should use an email address which you have access to so you can receive the two-factor authentication code – if using Gmail, you can add ‘+1’ onto the end of your email address to create additional accounts, if needed. For example, [janedoe+1234@example.com](mailto:janedoe+1234@example.com). If you are using another email provider, you might not be able to access this feature.
5. Enter the 6-digit security code sent to your email – it will have a subject line similar to ‘Your security code for your GOV.UK One Login’.
6. Create a password.
7. Select **Authenticator app for smartphone, tablet or computer**.
8. Select the **I cannot scan the QR code** dropdown.
9. Make a note of the secret key which appears in the dropdown – some authenticator apps call the secret key a ‘code’.
10. Use this secret key to generate a one-time code using a scripting language within your test – there’s an [example of how to generate a one-time code using TypeScript \(<https://github.com/govuk-one-login/onboarding-examples/blob/main/tools/totp/totp.ts>\)](#) in our GitHub repo.

# Conducting end-to-end user testing against the integration environment

## Test successful user journeys

Before you can test successful authentication or identity proving journeys, you need to:

1. Check you can connect to the integration environment.
2. [Contact GOV.UK One Login to access test user data](#) – you'll use this to test your journeys.

### Test a successful authentication journey

You should test if you can authenticate users successfully. This scenario uses a web-based journey to create a GOV.UK One Login.

1. Go to your service start page.
2. Select **Start**.
3. Select **Create a GOV.UK One Login**.
4. Follow the instructions to create an account using the test user data. You should use an email address which you have access to so you can receive the two-factor authentication code – if using Gmail, you can add '+1' onto the end of your email address to create additional accounts, if needed. For example, [janedoe+1234@example.com](mailto:janedoe+1234@example.com). If you are using another email provider, you might not be able to access this feature.
5. Enter the 6-digit security code sent to your email – it will have a subject line similar to 'Your security code for your GOV.UK One Login'.
6. Create a password.
7. Select how you want to receive your security codes.
8. Select **Continue**.

### Test a successful identity proving journey

If your service provides identity proving functionality, you should test if you can prove your users' identities successfully. This scenario uses a web-based journey to create a GOV.UK One Login.

1. Go to your service start page.
2. Select **Start**.
3. Select **Create a GOV.UK One Login**.
4. Follow the instructions to create an account using the test user data. You should use an email address which you have access to so you can receive the two-factor authentication code – if using Gmail, you can add '+1' onto the end of your email

address to create additional accounts, if needed. For example, [janedoe+1234@example.com](mailto:janedoe+1234@example.com). If you are using another email provider, you might not be able to access this feature.

5. Enter the 6-digit security code sent to your email – it will have a subject line similar to ‘Your security code for your GOV.UK One Login’.
6. Create a password.
7. Select how you want to receive your security codes.
8. Select **Continue**.
9. Select **Continue** when asked about proving your identity with GOV.UK One Login.
10. Select **Yes**, then **Continue** when asked if you have a photo ID.
11. Select **Yes, I am on a computer or tablet**, then **Continue**.
12. Select **I don't have either of these** when asked if you have a smartphone.
13. Select **UK photocard driving licence** or **UK passport** when asked if you want to use your UK photocard driving licence or UK passport to prove your identity, then **Continue**.
14. Fill in the document details from the test user data profiles, then **Continue**.
15. Enter the postcode from the test user data profiles.
16. Select **Find address**.
17. Find the correct address from the dropdown list and select **Choose address**.
18. Enter the correct year from the test user data profiles into **When did you start living here**, then **Continue**.
19. Select **I confirm my details are correct** then **Continue**.
20. Select **Continue**, and answer the security question from the test user data profiles (this will be in the knowledge-based verification question section in the test user data profiles document). You must answer 3 correctly and can only get a maximum of 1 wrong.
21. Select **Continue**.

## Test unsuccessful user journeys

You should test if your service recognises failed authentication or identity proving journeys. Before you can test these, you need to:

1. Check you can connect to the integration environment.
2. [Contact GOV.UK One Login to access test user data](#) – you’ll use this to test your journeys.

To test a failed journey, you need to input incorrect data. For example, inputting an incorrect date of birth, or document number.

## Test a failed identity proving journey

If your service provides identity proving functionality, you should test a failed identity proving journey.

Your test outcome will vary depending on whether you use the return code claim or not.

### Test a failed identity proving journey without the return code claim

1. Go to your service start page.
2. Select **Start**.
3. Select **Create a GOV.UK One Login**.
4. Follow the instructions to create an account using the test user data. You should use an email address which you have access to so you can receive the two-factor authentication code – if using Gmail, you can add ‘+1’ onto the end of your email address to create additional accounts, if needed. For example, [janedoe+1234@example.com](mailto:janedoe+1234@example.com). If you are using another email provider, you might not be able to access this feature.
5. Enter the 6-digit security code sent to your email – it will have a subject line similar to ‘Your security code for your GOV.UK One Login’.
6. Create a password.
7. Select how you want to receive your security codes, then **Continue**.
8. Select **Continue** when asked about proving your identity with GOV.UK One Login.
9. Select **Yes**, then **Continue** when asked if you have a photo ID.
10. Select **Yes, I am on a computer or tablet**, then **Continue**.
11. Select **I don't have either of these** when asked if you have a smartphone.
12. Select **UK photocard driving licence** or **UK passport** when asked if you want to use your UK photocard driving licence or UK passport to prove your identity, then **Continue**.
13. Fill in the document details from the test user data profiles but input incorrect data – for example, an incorrect date of birth, or document number, then **Continue**.
14. When you see the error message ‘Sorry, you’ll need to prove your identity another way’, select **Prove your identity another way**.
15. Select **Continue** and you’ll receive an OAuth ‘Access Denied’ error to your [redirect\\_uri](#).

### Test a failed identity proving journey using the return code claim

If you’re using the [return code claim \(<https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/prove-users-identity/#understand-your-user-s-return-code-claim>\)](https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/prove-users-identity/#understand-your-user-s-return-code-claim), you should test different ways of how an identity proving journey might fail. Your integration should receive the expected return code back, and handle it appropriately.

For example, submitting an incorrect document number will return an error which explains it was not possible to confirm a user’s identity.

1. Go to your service start page.
2. Select **Start**.
3. Select **Create a GOV.UK One Login**.

4. Follow the instructions to create an account using the test user data. You should use an email address which you have access to so you can receive the two-factor authentication code – if using Gmail, you can add ‘+1’ onto the end of your email address to create additional accounts, if needed. For example, [janedoe+1234@example.com](mailto:janedoe+1234@example.com). If you are using another email provider, you might not be able to access this feature.
5. Enter the 6-digit security code sent to your email – it will have a subject line similar to ‘Your security code for your GOV.UK One Login’.
6. Create a password.
7. Select how you want to receive your security codes, then **Continue**.
8. Select **Continue** when asked about proving your identity with GOV.UK One Login.
9. Select **Yes**, then **Continue** when asked if you have a photo ID.
10. Select **Yes, I am on a computer or tablet**, then **Continue**.
11. Select **I don't have either of these** when asked if you have a smartphone.
12. Select **UK photocard driving licence** or **UK passport** when asked if you want to use your UK photocard driving licence or UK passport to prove your identity, then **Continue**.
13. Fill in the document details from the test user data profiles but input incorrect data – for example, an incorrect date of birth, or document number, then **Continue**.
14. When you see the error message ‘Sorry, you’ll need to prove your identity another way’, select **Prove your identity another way**.
15. Select **Continue** and you’ll receive a `returnCode` in your response from `/userinfo` – there’s further guidance on return codes (<https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/prove-users-identity/#understand-your-user-s-return-code-claim>).

## Navigating internal performance testing of your service

For performance testing, you should focus on the processing and successful handling of the agreed request and response volumes back into your service.

You are responsible for conducting performance testing against your own system. You should use the GOV.UK One Login simulator to test your system (</test-your-integration/gov-uk-one-login-simulator/>) as GOV.UK One Login does not provide environments for this.

You must not:

- performance test any GOV.UK One Login environment
- use any GOV.UK One Login environment to do performance testing of your service

If GOV.UK One Login detects an unusual amount of requests from the same IP address, you may see errors. In extreme cases, GOV.UK One Login may block your IP address.

GOV.UK One Login is responsible for performance testing the agreed volumes of requests into the GOV.UK One Login service.

# Avoid penetration testing

You must not do any penetration ‘pen’ testing against GOV.UK One Login’s environment.

This page was last reviewed on 17 February 2025.

[View source](#) [Report problem](#) [GitHub Repo](#)



[Accessibility](#)

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Test your integration with GOV.UK One Login

Once you've [integrated your service with Authorization Code Flow \(/integrate-with-integration-environment/authenticate-your-user/\)](#), you can test your integration with GOV.UK One Login.

You have 2 options for testing:

- the [GOV.UK One Login simulator \(/test-your-integration/gov-uk-one-login-simulator/\)](#), which lets you test and verify specific user information and error codes
- using the [GOV.UK One Login integration environment \(/test-your-integration/using-integration-for-testing/\)](#), which lets you test end to end user journeys

## Compare GOV.UK One Login simulator and integration environments

The GOV.UK One Login simulator does not currently support all GOV.UK One Login features. Use this table to understand the difference between the GOV.UK One Login simulator and the integration environment.

| Feature                       | GOV.UK One Login simulator | GOV.UK One Login Integration environment   |
|-------------------------------|----------------------------|--|
| Uses the GOV.UK One Login API | Yes                        | Yes  |
| Configurable response data    | Yes                        | No. You need to request fictional users and their knowledge-based verification (KBV) answers to help you test your journeys. <a href="#">Email GOV.UK One Login</a> to access this test user data. |

|   |   |     |
|---|---|-----|
| Supports <a href="#">client_secret_post</a>       | No  | Yes |
| Runs on a publicly accessible endpoint            | No, unless you host it online.  | Yes |
| Runs locally                                      | Yes   | No  |
| Supports permit missing nonce                     | No  | Yes |
| Configure with the GOV.UK self service admin tool | No. There's further guidance about <a href="#">configuring the GOV.UK One Login simulator (/test-your-integration/gov-uk-one-login-simulator/#configure-the-gov-uk-one-login-simulator)</a> for more information. | Yes |
| Supports performance testing                      | Yes - in <a href="#">interactive mode (/test-your-integration/gov-uk-one-login-simulator/#returning-multiple-response-configurations)</a> only.   | No  |
| Can test error messages                           | Yes   | No  |
| Can test the web journey                          | Yes   | Yes |
| Can test the mobile journey                       | No  | No  |
| Can test the landing page URL                     | No  | No  |



## Accessibility

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# GOV.UK One Login

GOV.UK One Login is the way for government services to:

- sign in their users
- prove their users' identity

This technical documentation gives you information on how to:

- plan the functionality your service needs
- register your service with GOV.UK One Login
- integrate with GOV.UK One Login to authenticate users and prove their identity
- configure your service for production

You can [read further documentation about how GOV.UK One Login works \(/how-gov-uk-one-login-works/\)](#).

Contact us if you have any questions on our [#govuk-one-login Slack channel \(https://ukgovernmentdigital.slack.com/archives/C02AQUJ6WTC\)](#).

## Documentation updates

These are the most recent changes to this documentation.

| Publication date | Update |
|------------------|--------|
|------------------|--------|

---

|             |  |
|-------------|--|
| Oct 23 2025 | Added guidance <a href="#">“Setting a User-Agent header on HTTP requests” (/before-integrating/set-user-agent-header)</a> the requirement to use an appropriate <b>User-Agent</b> header on service calls to GOV.UK One Login. |
|-------------|--|

---

|            |   |
|------------|---|
| Sep 2 2025 | Updates guidance <a href="#">“Prove your user’s identity” (/integrate-with-integration-environment/prove-users-identity/#prove-your-user-39-s-identity)</a> with guidance for helping your users after their in-person identity checks. |
|------------|---|

|             |  |
|-------------|--|
| Jul 30 2025 | Updates guidance ‘ <a href="#">Choose which scopes your service can request</a> ’ ( <a href="#">/before-integrating/choose-which-user-attributes-your-service-can-request/#choose-which-scopes-your-service-can-request</a> ) and ‘ <a href="#">Retrieve user information</a> ’ ( <a href="#">/integrate-with-integration-environment/authenticate-your-user/#receive-response-for-retrieve-user-information</a> ) to add information about the <b>wallet-subject-id</b> scope.  |
| Jun 12 2025 | Updates section on testing to remove guidance on building mocks and move guidance on “ <a href="#">using the GOV.UK One Login simulator</a> ” ( <a href="#">/test-your-integration/gov-uk-one-login-simulator/</a> ) to section on “ <a href="#">testing your integration with GOV.UK One Login</a> ” ( <a href="#">/test-your-integration/</a> ).   |
| May 2 2024  | Updates guidance to add information about using <a href="#">Proof Key for Code Exchange (PKCE)</a> parameters in the authorise request. ( <a href="#">/integrate-with-integration-environment/authenticate-your-user/#make-a-request-to-the-authorize-endpoint</a> ) Updates guidance <a href="#">to include information about using PKCE parameters in the token request</a> . ( <a href="#">/integrate-with-integration-environment/authenticate-your-user/#make-a-token-request</a> ) Updates guidance <a href="#">to include guidance about PKCEEnforced field when configuring your service</a> ( <a href="#">/configure-for-production/</a> ). |
| Apr 15 2025 | Updates guidance “ <a href="#">Managing user sessions if your service session is less than 1 hour</a> ” ( <a href="#">/integrate-with-integration-environment/managing-your-users-sessions/#managing-user-sessions-if-your-service-session-is-less-than-1-hour</a> ) to add guidance on how to re-authenticate your users. Updates the ‘ <a href="#">Make a request to the /authorize endpoint</a> ’ table ( <a href="#">/integrate-with-integration-environment/authenticate-your-user/#error-handling-for-make-a-request-to-the-authorize-endpoint</a> ) to add an entry for <b>login_required</b> error code.                                     |
| Apr 2 2025  | New guidance “ <a href="#">Test your service with the GOV.UK One Login simulator</a> ” ( <a href="#">/test-your-integration/gov-uk-one-login-simulator/</a> ) to add information about the new GOV.UK One Login simulator.   |
| Mar 5 2025  | Updates guidance “ <a href="#">Integrating third-party platforms with GOV.UK One Login</a> ” ( <a href="#">/before-integrating/integrating-third-party-platform/#integrating-third-party-platforms-with-gov-uk-one-login</a> ) to add guidance on integrating with GOV.UK One Login using Amazon Cognito.  |
| Feb 17 2025 | Updates guidance “ <a href="#">Using the integration environment for end-to-end testing</a> ” ( <a href="#">/test-your-integration/using-integration-for-testing/#using-the-integration-environment-for-end-to-end-testing</a> ) to remove reference to the integration environment basic authentication challenge which has been removed and is no longer required.   |
| Jan 27 2025 | Updates guidance “ <a href="#">Authenticate your user</a> ” ( <a href="#">/integrate-with-integration-environment/authenticate-your-user</a> ) to add information about using the  |

`max_age` parameter. Updates guidance “[Generate an authorisation code](#)” ([/integrate-with-integration-environment/authenticate-your-user](#)) to add information about validating `max_age` parameter.

|             |   |
|-------------|---|
| Jan 21 2025 | New guidance “ <a href="#">Quick start</a> ” ( <a href="#">/quick-start/</a> ) to help users see how a typical integration with GOV.UK One Login works.   |
| Oct 23 2024 | Updates guidance “ <a href="#">Understand the core identity signing key rotations</a> ” ( <a href="#">/integrate-with-integration-environment/prove-users-identity/#understand-the-core-identity-signing-key-rotations</a> ) to add information on the frequency of key rotations for the environments.   |
| Oct 22 2024 | Updates and renames ‘Generate a key pair’ page to include new guidance “ <a href="#">share your public keys using a JWKS endpoint</a> ” to add other option when sharing your public key with GOV.UK One Login.   |
| Sep 25 2024 | Updates guidance “ <a href="#">Register and manage your service</a> ” ( <a href="#">/before-integrating/register-and-manage-your-service/#register-and-manage-your-service</a> ) to add guidance on how to register and manage a service.   |
| Sep 17 2024 | Updates guidance “ <a href="#">Integrating third-party platforms with GOV.UK One Login</a> ” ( <a href="#">/before-integrating/integrating-third-party-platform/#integrating-third-party-platforms-with-gov-uk-one-login</a> ) to add guidance on integrating with GOV.UK One Login using Salesforce.   |
| Sep 6 2024  | Updates guidance “ <a href="#">Use the production discovery endpoint</a> ” ( <a href="#">/configure-for-production/#use-the-production-discovery-endpoint</a> ) to add the production discovery endpoint.   |
| Aug 21 2024 | Updates guidance “ <a href="#">Configure your service for production</a> ” ( <a href="#">/configure-for-production/</a> ) to add information about how to configure your service for production.  |
| Aug 20 2024 | Updates guidance “ <a href="#">Receive response for ‘Retrieve user information’</a> ” ( <a href="#">/integrate-with-integration-environment/authenticate-your-user/#receive-response-for-retrieve-user-information</a> ) to add a table explaining more about the response from the <code>/userinfo</code> endpoint.                            |
| Jul 29 2024 | Updates guidance “ <a href="#">Error handling for ‘Make a request to the /authorize endpoint’</a> ” ( <a href="#">/integrate-with-integration-environment/authenticate-your-user/#error-handling-for-make-a-request-to-the-authorize-endpoint</a> ) to update we now return HTTP 400 Bad Request errors for requests with incorrect parameters. |
| Jul 18 2024 | New guidance “ <a href="#">Validate the core identity claim JWT using a public key</a> ” ( <a href="#">/integrate-with-integration-environment/prove-users-identity/#validate-the-core-identity-claim-jwt-using-a-public-key</a> ). Contains information about validating   |

the core identity claim JWT using a public key, which GOV.UK One Login publishes in its Decentralized Identifier (DID) documents.

---

|             |   |
|-------------|---|
| Jul 9 2024  | Removes the <a href="https://vocab.account.gov.uk/v1/socialSecurityRecord">https://vocab.account.gov.uk/v1/socialSecurityRecord</a> claim   |
| Jul 4 2024  | New guidance <a href="#">“Integrating third-party platforms with GOV.UK One Login” (/before-integrating/integrating-third-party-platform/)</a> which contains information about integrating with GOV.UK One Login using a third-party platform, and contains details about the <code>client_secret_post</code> token authentication method. |
| Jun 21 2024 | Updates guidance <a href="#">“Error handling for ‘Make a request to the /authorize endpoint” (/integrate-with-integration-environment/authenticate-your-user/#error-handling-for-make-a-request-to-the-authorize-endpoint)</a> to clarify the <code>{"message": "Internal server error"}</code> HTTP 502 Bad gateway error.                 |
| Jun 18 2024 | Includes example data to help with building mocks: Access example data.   |
| May 22 2024 | New guidance <a href="#">Using the integration environment for end-to-end testing (/test-your-integration/using-integration-for-testing/)</a> to explain how to use the integration environment for end-to-end testing.   |
| May 17 2024 | New guidance Build mocks to work with GOV.UK One Login to explain how to build mocks as a part of testing your service.   |
| May 2 2024  | New guidance <a href="#">Managing your users’ sessions (/integrate-with-integration-environment/managing-your-users-sessions/)</a> to explain how to manage your users’ sessions and how to build a logout mechanism for your users.  |
| Apr 9 2024  | Updates the <a href="#">technical flow diagram (/how-gov-uk-one-login-works/#understand-the-technical-flow-gov-uk-one-login-uses)</a> to document the use of the <code>/logout</code> endpoint.   |
| Apr 3 2024  | New guidance <a href="#">Understand your user’s return code claim (/integrate-with-integration-environment/prove-users-identity/#understand-your-user-s-return-code-claim)</a> which gives information about any issues with the evidence your user provided to prove their identity.   |
| Mar 25 2024 | Removes references to the refresh token and <code>offline_access</code> to simplify integration and the technical flow.   |
| Feb 14 2024 | New guidance <a href="#">Choose your sector identifier (/before-integrating/choose-your-sector-identifier/)</a> to explain the use of the sector identifier with a  |

worked example that shows the effect of choosing different sector identifiers.

---

|                |   |
|----------------|---|
| Dec 22<br>2023 | Updates guidance on making a request to the <code>/authorize</code> endpoint.   |
| Dec 21<br>2023 | New guidance <a href="#">Secure your authorisation request parameters with JWT (/integrate-with-integration-environment/authenticate-your-user/#secure-your-authorisation-request-parameters-with-jwt)</a> using a JWT-secured OAuth 2.0 authorisation request (JAR) to improve the security of your integration and protect against tampering. |
| Oct 31 2023    | New guidance <a href="#">Before you integrate with GOV.UK One Login (/before-integrating)</a> .   |

---

This page was last reviewed on 17 September 2024.

[View source](#) [Report problem](#) [GitHub Repo](#)



[Accessibility](#)

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

© Crown copyright

[Table of contents](#)

# GOV.UK Wallet tech docs accessibility statement

## Accessibility statement for GOV.UK Wallet technical documentation

This accessibility statement applies to the GOV.UK Wallet technical documentation at <https://docs.wallet.service.gov.uk/> (<https://docs.wallet.service.gov.uk/>).

This website is run by the GOV.UK One Login team at the Government Digital Service (GDS). We want as many people as possible to be able to use this website. For example, that means you should be able to:

- change colours, contrast levels and fonts
- zoom in up to 300% without problems
- navigate most of the website using just a keyboard
- navigate most of the website using speech recognition software
- listen to most of the website using a screen reader (including the most recent versions of JAWS, NVDA and VoiceOver)

We've also made the website text as simple as possible to understand.

[AbilityNet](https://abilitynet.org.uk/) (<https://abilitynet.org.uk/>) has advice on making your device easier to use if you have a disability.

### How accessible this website is

This website is partially compliant with the Web Content Accessibility Guidelines version 2.2 AA standard.

### What to do if you cannot access parts of this website

If you need information on this website in a different format like accessible PDF, large print, easy read, audio recording or braille, [use the Contact Us page to contact the](#)

[GOV.UK Wallet team \(/contact-us.html\)](/contact-us.html) with details of your request.

We'll aim to reply in 3 working days.

## Reporting accessibility problems with this website

We're always looking to improve the accessibility of this website. If you find any problems not listed on this page or think we're not meeting accessibility requirements, [use the Contact Us page to contact the GOV.UK Wallet team \(/contact-us.html\)](/contact-us.html).

## Enforcement procedure

The Equality and Human Rights Commission (EHRC) is responsible for enforcing the Public Sector Bodies (Websites and Mobile Applications) (No. 2) Accessibility Regulations 2018 (the 'accessibility regulations'). If you're not happy with how we respond to your complaint, [contact the Equality Advisory and Support Service \(EASS\) \(https://www.equalityadvisoryservice.com/\)](https://www.equalityadvisoryservice.com/).

## Technical information about this website's accessibility

GDS is committed to making its website accessible, in accordance with the Public Sector Bodies (Websites and Mobile Applications) (No. 2) Accessibility Regulations 2018.

## Compliance status

This website is partially compliant with the Web Content Accessibility Guidelines version 2.2 AA standard.

## Preparation of this accessibility statement

This statement was prepared on 06 May 2025.

This website was last tested in May 2025.

This page was last reviewed on 6 May 2025. It needs to be reviewed again on 6 November 2025 .



## Accessibility

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Before you issue a credential

Before you can issue a credential, you should do the following.

## Onboard with GOV.UK One Login

GOV.UK Wallet uses GOV.UK One Login to authenticate users. This process makes sure that credentials are issued into a wallet that is logged in as the same user the credential is for.

You must complete the onboarding process for GOV.UK One Login before you can issue credentials to GOV.UK Wallet. There is [guidance on the GOV.UK One Login onboarding process in their technical documentation \(<https://docs.sign-in.service.gov.uk/>\)](#).

When you complete the GOV.UK One Login onboarding process, make sure that you have:

- your unique client identifier - this identifier must be included as a claim (`client_id`) in the pre-authorised code your service generates as part of issuing a GOV.UK Wallet credential offer
- requested that the `walletSubjectId` custom claim is activated for your GOV.UK One Login client - this is a pairwise identifier that will be used to prove that the user logged in to your service and GOV.UK Wallet are the same user

## Agree a credential template

Before you can issue a credential, you must confirm with the GOV.UK Wallet onboarding team:

- the credential type - this can be [mDoc \(<https://www.iso.org/obp/ui/en/#iso:std:iso-iec:18013:-5:ed-1:v1:en>\)](#) or [JWT VC \(<https://datatracker.ietf.org/doc/draft-ietf-oauth-sd-jwt-vc/>\)](#)
- which attributes are to be included in the digital credential and its schema
- the colour of the digital card
- whether users can have multiple instances of this credential, or only one at a time

To integrate with GOV.UK Wallet you must send the GOV.UK Wallet team:

- your issuer URL (both integration and production)
- your issuer logo in English and Welsh

## Follow relevant standards

When you are preparing to issue a credential to GOV.UK Wallet you must align with open standards.

GOV.UK Wallet will support multiple credential formats to represent government documents. These documents can be:

- [mdoc based credentials for the digital driving licence](https://www.iso.org/obp/ui/en/#iso:std:iso-iec:18013:-5:ed-1:v1:en) (<https://www.iso.org/obp/ui/en/#iso:std:iso-iec:18013:-5:ed-1:v1:en>)
- other Verifiable Credentials (VCs), including [W3C Verifiable Credential Data Model 2.0](https://www.w3.org/TR/vc-data-model-2.0/) (<https://www.w3.org/TR/vc-data-model-2.0/>) and later [other formats allowing selective disclosure of attributes](https://datatracker.ietf.org/doc/draft-ietf-oauth-sd-jwt-vc/) (<https://datatracker.ietf.org/doc/draft-ietf-oauth-sd-jwt-vc/>)

GOV.UK Wallet supports [OpenID Connect for Verifiable Credential Issuance \(OIDC4VCI\)](https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html) ([https://openid.net/specs/openid-4-verifiable-credential-issuance-1\\_0.html](https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html)) for its issuance flow.

You should also:

- follow the government [Service Standard](https://www.gov.uk/service-manual/service-standard) (<https://www.gov.uk/service-manual/service-standard>), for example making all information available in Welsh
- use findings identified by your user research to meet your users' needs

To get started, you need to [authenticate users with One Login](#).

This page was set to be reviewed before 15 October 2025. This might mean the content is out of date.





[Table of contents](#)

# Credentials in GOV.UK Wallet

Credentials are issued by the relevant government department into GOV.UK Wallet. The government department defines which attributes to include by using the credential schema. They are also responsible for the information in the credentials they issue.

Credentials and their attributes will be made available in this technical documentation as they are released.

## Currently available credentials

### Veteran card

| Attribute | Definition  | Example  |
|-----------|---|--|
| Name      | Given name(s) and family names as they appear on the veteran card | <pre>"name": [   {     "nameParts": [       {         "value": "Sarah",         "type": "GivenName"       },       {         "value": "Elizabeth",         "type": "GivenName"       },       {         "value": "Edwards",         "type": "FamilyName"       }     ]   } ]</pre> |

Date of birth Day, month and year of birth as it appears on the veteran card

```
"birthDate": [  
  {  
    "value": "1985-10-18"  
  }  
]
```

|                        |  |   |
|------------------------|--|---|
| Photo                  | Photo of the cardholder  | "photo": "[PHOTO]"                          |
| Service number         | Identification code of the veteran card holder                 | "serviceNumber": "25057386"                 |
| Service branch         | Branch of the British Armed Forces where the cardholder served | "serviceBranch": "British Army"             |
| Expiry date            | Expiry date of the veteran card                                | "expiryDate": "2034-04-08"                  |
| Credential expiry date | Expiry date of the digital veteran card credential             | "exp": 1778664693<br>(13 May 2026 09:31:33) |

This page was last reviewed on 7 May 2025. It needs to be reviewed again on 7 November 2025 .



## Accessibility

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Supported formats and protocols

GOV.UK Wallet will support multiple credential formats to represent government documents. These documents can be:

- [mdoc based credentials for the digital driving licence](https://www.iso.org/obp/ui/en/#iso:std:iso-iec:18013:-5:ed-1:v1:en)  
(<https://www.iso.org/obp/ui/en/#iso:std:iso-iec:18013:-5:ed-1:v1:en>)
- other Verifiable Credentials (VCs), including [W3C Verifiable Credential Data Model 2.0](https://www.w3.org/TR/vc-data-model-2.0/) (<https://www.w3.org/TR/vc-data-model-2.0/>) and [other formats allowing selective disclosure of attributes](https://datatracker.ietf.org/doc/draft-ietf-oauth-sd-jwt-vc/) (<https://datatracker.ietf.org/doc/draft-ietf-oauth-sd-jwt-vc/>)

It will also support multiple protocols for users to share credentials and attributes:

- [OpenID for Verifiable Presentation \(OID4VP\)](https://openid.net/specs/openid-4-verifiable-presentations-1_0.html) ([https://openid.net/specs/openid-4-verifiable-presentations-1\\_0.html](https://openid.net/specs/openid-4-verifiable-presentations-1_0.html)) for online sharing, using remote flows to verify mdoc and VC based credentials
- [ISO/IEC 18013-5](https://www.iso.org/obp/ui/en/#iso:std:iso-iec:18013:-5:ed-1:v1:en) (<https://www.iso.org/obp/ui/en/#iso:std:iso-iec:18013:-5:ed-1:v1:en>) for in-person interactions, using proximity flows to verify mdoc based credentials

GOV.UK Wallet will first enable supervised proximity flow for verification in line with ISO/IEC 18013-5. Remote flows, and more details about OID4VP profiles, will be added in the future.

This page was last reviewed on 7 May 2025. It needs to be reviewed again on 7 November 2025 .





[Table of contents](#)

# Consuming and verifying credentials in GOV.UK Wallet

GOV.UK Wallet will allow GOV.UK One Login users to store and present digital versions of government-issued documents on their phones.

Government departments will [issue cryptographically verifiable credentials to a user's GOV.UK Wallet \(/issuing-credentials-to-wallet.html\)](#). The user's credentials are linked to their GOV.UK One Login account and to their personal device, and cannot be moved to another device.

GOV.UK Wallet will enable:

- government departments and certain other public sector organisations to consume and verify credentials or attributes
- organisations outside of government to use verified information passed to them by a digital verification service (DVS), certified under the [UK digital identity and attributes trust framework \(<https://www.gov.uk/government/collections/uk-digital-identity-and-attributes-trust-framework>\)](#) and on the [digital identity and attribute services register \(<https://www.gov.uk/guidance/find-registered-digital-identity-and-attribute-services>\)](#) (DVS register)

This technical documentation will be updated as new information and features are available. We welcome feedback from partners and industry on our documentation - find out how to [contact us \(/contact-us.html\)](#).

## Sharing credentials using GOV.UK Wallet

GOV.UK Wallet will let users share their credentials:

- in-person, for example to prove their age when purchasing age-restricted products
- online, to share documents with a service securely instead of uploading a photo or a PDF

GOV.UK Wallet will use [standard protocols to offer flexible verification scenarios \(/consuming-and-verifying-credentials/supported-protocols.html\)](#).

# Trusted list

GOV.UK Wallet will put mechanisms in place to make sure personal data from users is shared only with trusted parties. This will mitigate the risk of malicious apps or services accessing credential data without the user's knowledge.

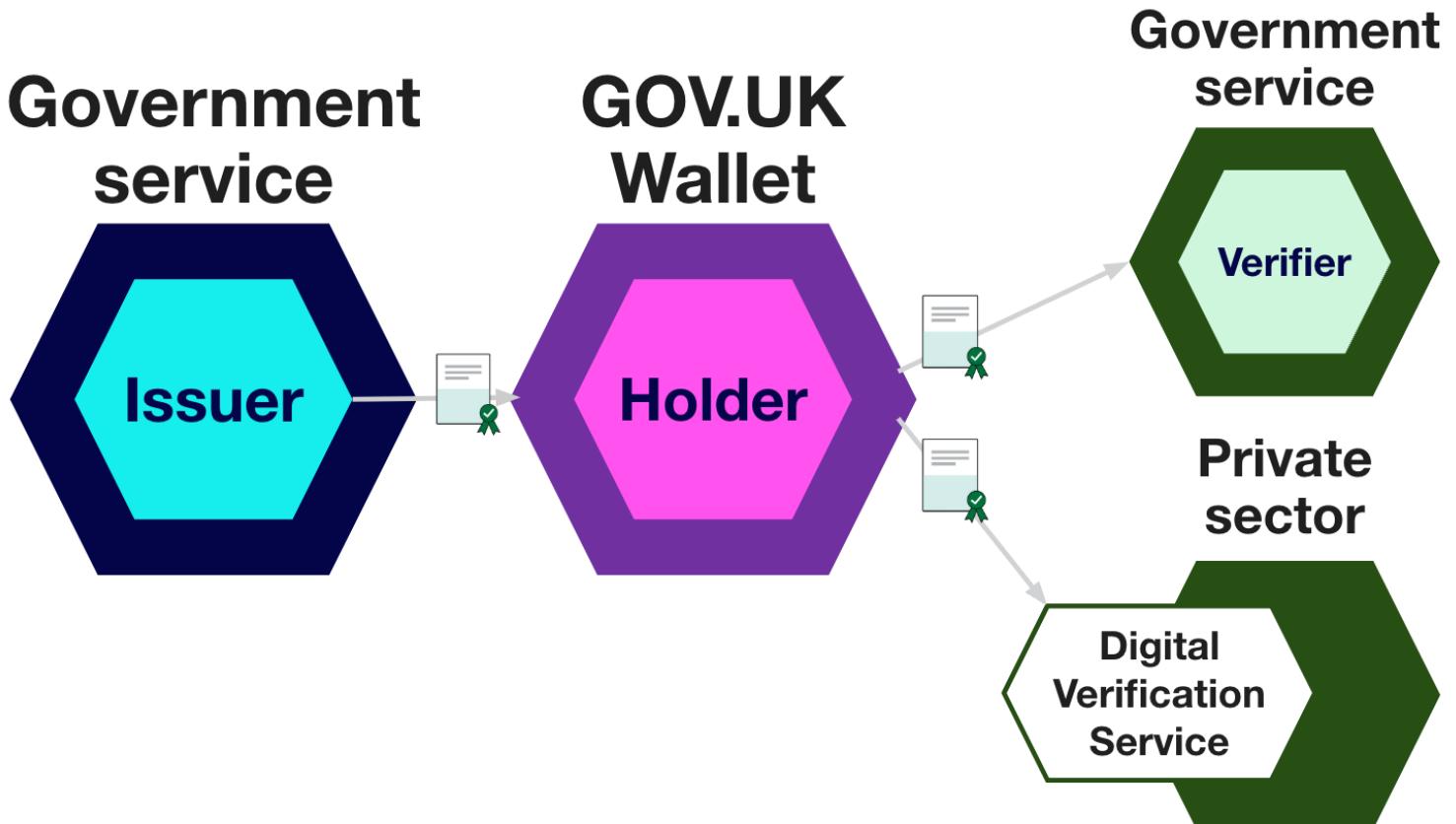
GOV.UK Wallet will use trusted lists to identify consumers of credentials. Where data is shared with parties outside of government, GOV.UK Wallet will only release credentials and attributes to a digital verification service (DVS) which is certified against the trust framework and appears on the DVS register.

Further details on this functionality will be added in future.

## Data flows

### Data flow between credential issuers, holders and verifiers

GOV.UK Wallet is built in three parts to connect government departments (credential issuers), users (credential holders) and verifiers requesting data (credential verifiers).



#### 1. Government department issuers

Government departments (issuers) issue digital and verifiable versions of physical documents (credentials) to a user's GOV.UK Wallet. For example, a government department could issue a credential containing a user's date of birth that proves their age.

## 2. GOV.UK Wallet

The credential's rightful holder (the user the credential refers to) uses GOV.UK Wallet to store, manage and present their credentials online and in person. For example, a user could store a credential containing their date of birth, and present information from it when they need to prove their age.

## 3. Verifier services

Government departments and certain public sector organisations will be able to verify and use credentials and attributes held in GOV.UK Wallet.

Outside of government, a DVS certified against the trust framework and added to the DVS register will be able to access GOV.UK Wallet and verify information it holds at a user's request.

For example, a business (known as a relying party) selling age-restricted products could use a certified and registered DVS to request and digitally verify a customer's age based on attributes held in credentials in their GOV.UK Wallet.

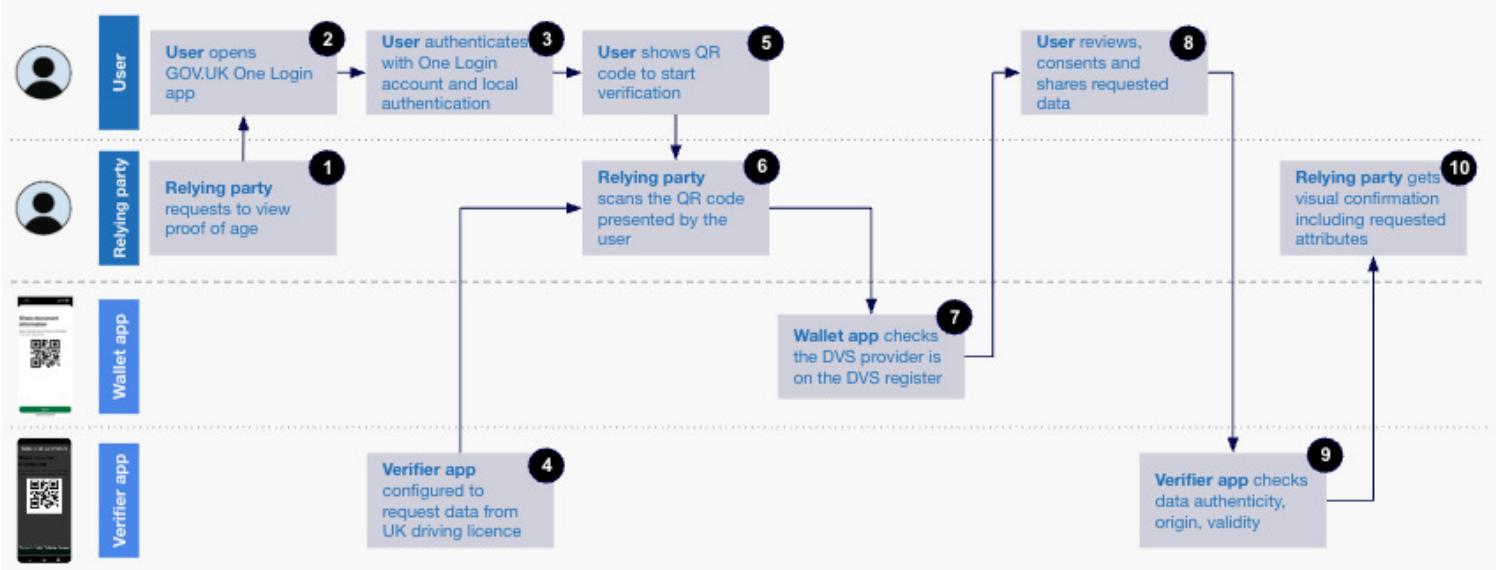
# Using GOV.UK Wallet in person in the private sector

The three example flows below illustrate how GOV.UK Wallet, via a registered DVS provider, can be used to purchase age-restricted products from a private sector business. All three assume the DVS is providing an orchestration service, but other models will also exist. There is [guidance on the models available for DVS providers](#) (<https://gov.uk/guidance/using-govuk-wallet-in-the-digital-identity-sector>).

## ISO 18013-5 supervised proximity flow

In this example, a user is purchasing age-restricted products in person and sharing their information with a business. They have a valid digital driving licence stored on their personal device in GOV.UK Wallet.

The business selling the products (the relying party) uses a device with a verification service provided by a suitable DVS to verify the user's age. In this example, the verification service is provided via a verifier app (it could, for example, also be a point of sale terminal, QR scanner terminal etc.). To work with GOV.UK Wallet, the DVS must be certified against the trust framework and appear on the DVS register.



The data flow for this interaction is as follows.

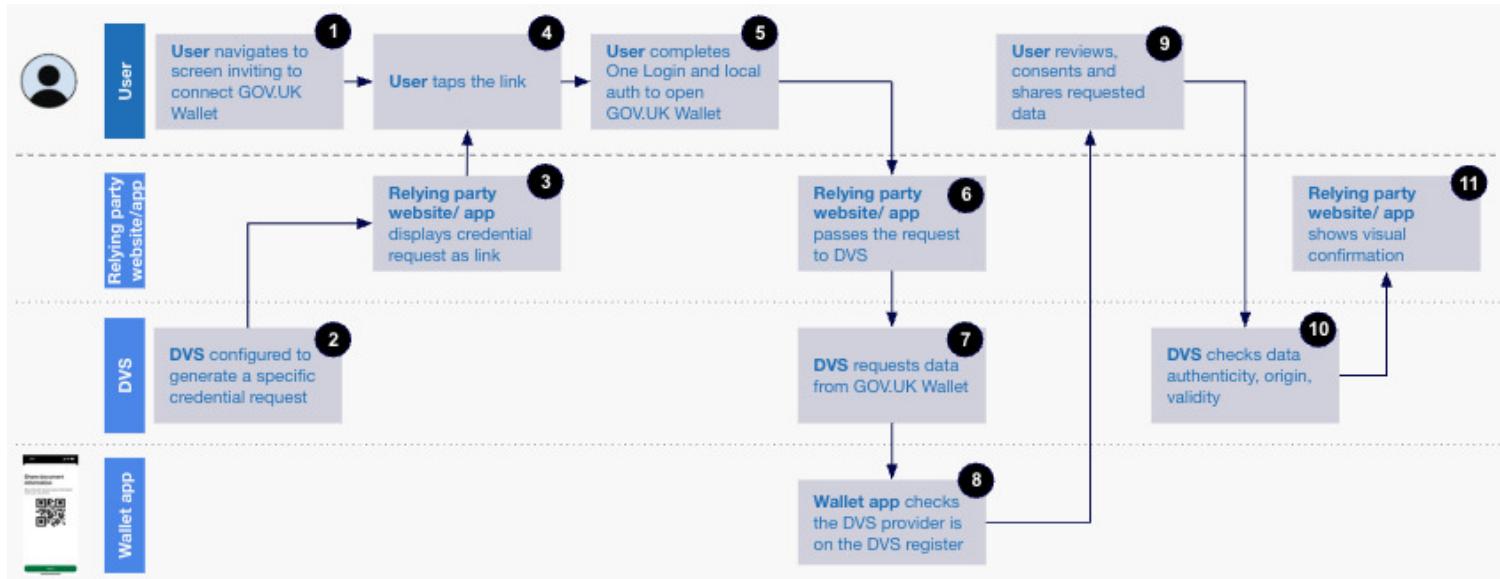
1. The relying party asks the user to show proof of their age.
2. The user who needs to prove their age opens the GOV.UK One Login app.
3. To open GOV.UK Wallet, the user authenticates themselves with GOV.UK One Login and uses the device's local authentication (face, fingerprint, PIN or pattern).
4. The verifier app on the relying party's device is configured to request data from the user's digital driving licence. For this transaction, the data requested is a proof of age.
5. GOV.UK Wallet generates a QR code on the user's device, which the user shows to the relying party to begin the verification process.
6. The relying party scans the QR code using the verifier app.
7. GOV.UK Wallet checks that the verifier app is using a trust framework certified and DVS-registered provider.
8. The user reviews the data that was requested (for example an 'over 18' attribute), consents to share it, and allows it to be shared with the verifier app.
9. The verifier app checks the data's authenticity, origin and validity.
10. The verifier app shows the relying party a visual confirmation of the user's proof of age.

## Using GOV.UK Wallet online in the private sector

### OID4VP same device flow

In this example, a user is purchasing an age-restricted product online using an app or the browser on their phone. This is the same phone where their credentials are held in GOV.UK Wallet. The user holds a credential that would prove their age (for example, a digital driving licence) in GOV.UK Wallet.

The website selling the product (the relying party) must get proof of the user's age before completing the transaction. To work with GOV.UK Wallet, the relying party website must use a registered DVS certified against the trust framework.



The data flow for this interaction is as follows.

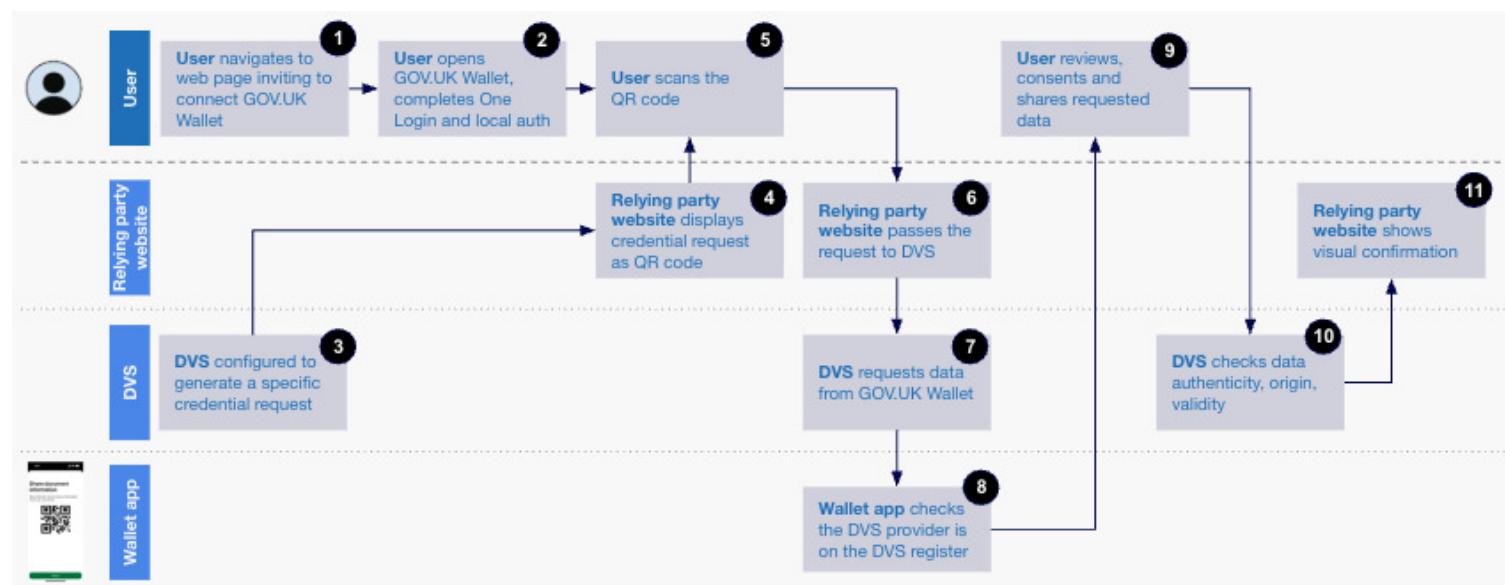
1. During their purchase, the relying party directs the user to a DVS to prove their age. The user chooses to connect this DVS to their GOV.UK Wallet.
2. The DVS used by the relying party's website is configured to generate a specific credential request. In this example, the credential request would include confirmation that the user is above the age needed to complete the transaction.
3. The relying party's website or embedded DVS displays the credential request to the user as a link.
4. The user taps the link, which opens the GOV.UK One Login app.
5. To open GOV.UK Wallet, the user authenticates themselves with GOV.UK One Login and uses the device's local authentication (face, fingerprint, PIN or pattern).
6. The relying party passes the request to the DVS.
7. The DVS requests the data it needs from GOV.UK Wallet. In this example, the data would be a confirmation that the user is above the age needed to complete the transaction.
8. GOV.UK Wallet checks that the DVS provider is on the DVS register.
9. The user reviews the data that was requested, consents to share it, and allows it to be shared with the DVS and the relying party.
10. The DVS checks the data's authenticity, origin and validity, and passes it to the relying party.
11. The relying party website/app shows visual confirmation.

11. The relying party website shows a visual confirmation of the credential verification. If the verification was successful and the user has proven their age, they can continue with the transaction.

## OID4VP cross device flow

In this example, the user holds a credential that would prove their age (for example, a digital driving licence) in GOV.UK Wallet on their phone. The user is purchasing an age-restricted product online using a separate device (for example, a laptop or tablet).

The website selling the product (the relying party) must get proof of the user's age before completing the transaction. The relying party is using a registered DVS certified against the trust framework.



The data flow for this interaction is as follows.

1. During their purchase on their laptop or tablet, the relying party directs the user to a DVS to prove their age. The user chooses to connect this DVS to their GOV.UK Wallet app on their phone.
2. To open GOV.UK Wallet on their phone, the user authenticates themselves with GOV.UK One Login and uses the device's local authentication (face, fingerprint, PIN or pattern).
3. The DVS used by the relying party is configured to generate a specific credential request. In this example, the request would include a confirmation that the user is above the age needed to complete the transaction.
4. The relying party displays the credential request to the user as a QR code.
5. The user scans the QR code using GOV.UK Wallet on their phone.
6. The relying party passes the request to the DVS.

7. The DVS requests the data it needs from GOV.UK Wallet. In this example, the data would be a confirmation that the user is above the age needed to complete the transaction.
8. GOV.UK Wallet checks that the DVS provider appears on the DVS register.
9. The user reviews the data that was requested, consents to share it, and allows it to be shared with the DVS and the relying party.
10. The DVS checks the data's authenticity, origin and validity, and passes it to the relying party
11. The relying party shows a visual confirmation of the credential verification. If verification was successful and the user has proven their age, they can continue with their purchase.

This page was last reviewed on 7 May 2025. It needs to be reviewed again on 7 November 2025 .



## Accessibility

## **OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Contact us

## For government and the public sector

If you have feedback or questions, contact us via the [#govuk-wallet Slack channel](#) (<https://ukgovernmentdigital.slack.com/archives/C08A9JMDK0Q>).

## For a digital verification service

If you are a digital verification service and have questions about using GOV.UK Wallet, or this documentation, email [govukwallet-queries@digital.cabinet-office.gov.uk](mailto:govukwallet-queries@digital.cabinet-office.gov.uk).

This page was last reviewed on 9 May 2025. It needs to be reviewed again on 9 November 2025 .



[Accessibility](#)

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Authenticating users with One Login

Services that wish to issue credentials must use GOV.UK One Login to authenticate their users. This process makes sure that credentials are issued into a wallet that is logged in as the same user the credential is for.

When you register your service with GOV.UK One Login, you get a unique client identifier. This identifier must be included as a claim (`client_id`) in the pre-authorised code your service generates as part of issuing a credential offer. There is more guidance on [issuing a credential offer \(/credential-issuer-functionality/credential-offer\)](#).

When your user authenticates with GOV.UK One Login, you obtain their user information, which includes their GOV.UK Wallet subject identifier (`walletSubjectId`). This subject identifier is a pairwise identifier you can use at the point where you finally issue the digital credential to assure that the user logged in to your service and GOV.UK Wallet are the same user. This is referred to as the ‘rightful holder check’.

This page was set to be reviewed before 5 September 2025. This might mean the content is out of date.

[Accessibility](#)**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# API

## /credential

### post

An endpoint used by the GOV.UK Wallet to request a Credential from the Credential Issuer.

#### Responses

| Status | Description |
|--------|-------------|
|--------|-------------|

|     |             |
|-----|-------------|
| 200 | Credential. |
|-----|-------------|

```
{  
  "credentials": [  
    {  
      "credential": "eyJraWQiOiJkaWQ6d2ViOmV4YW1wbGUtY3JlZGVudGlhbC1pc3N1ZXIuZ292LnVrIzVkJ2JlZTg2M2I1ZDdjYzMwYzliYT...  
    }  
  ],  
  "notification_id": "776aefd4-26c6-4a5f-aa7c-b5e294cd87cd"  
}
```

|     |             |
|-----|-------------|
| 400 | Bad Request |
|-----|-------------|

```
{  
  "error": "invalid_proof"  
}
```

|     |              |
|-----|--------------|
| 401 | Unauthorized |
|-----|--------------|

## Schemas

### CredentialResponse

| Name            | Type   | Required | Description                                | Schema                     |
|-----------------|--------|----------|--|----------------------------|
| credentials     | array  | false    | An array containing one issued credential. | <a href="#">Credential</a> |
| notification_id | string | false    | Issuance flow notification ID.             |                            |

### Credential

| Name       | Type   | Required | Description            | Schema |
|------------|--------|----------|------------------------|--------|
| credential | string | true     | The issued credential. |        |

### Credential400ErrorResponse

| Name  | Type   | Required | Description   | Schema |
|-------|--------|----------|---|--------|
| error | string | false    | An error code, such as <code>invalid_proof</code> or <code>invalid_nonce</code> . |        |

This page was last reviewed on 19 June 2025. It needs to be reviewed again on 19 December 2025.



## Accessibility

**OGL** All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# API

## /credential\_offer

### get

An endpoint for issuing a credential offer.

#### Responses

| Status | Description |
|--------|-------------|
|--------|-------------|

|     |   |
|-----|---|
| 200 | Credential offer URL consisting of the following parts: |
|-----|---|

1. Wallet's credential offer endpoint.
2. A query parameter `credential_offer`.
3. URL-encoded credential offer.

```
"https://mobile.account.gov.uk/wallet/add?credential_offer=%7B%22credential_configuration_ids%22%3A%5B%22FishingLic
```

This page was set to be reviewed before 24 September 2025. This might mean the content is out of date.



#### [Accessibility](#)

**OGL** All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

© Crown copyright

[Table of contents](#)

# Credential Offer

A credential offer is used to pass information relevant for credential issuance to GOV.UK Wallet, including a unique credential identifier so the credential can be identified later. Your service must generate a credential offer to begin the credential issuance process. This credential offer is passed to GOV.UK Wallet after an authenticated user gives their consent.

You must build your credential offer using the full `credential_offer` object embedded in a URI. GOV.UK Wallet does not support using the `credential_offer_uri` parameter to reference a JSON object containing credential offer parameters. There is more guidance in the [OID4VCI specification \(`https://openid.net/specs/openid-4-verifiable-credential-issuance-1\_0.html#name-credential-offer`\)](https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html#name-credential-offer).

## Technical details

### The credential offer object

The credential offer is a JSON object containing the following parameters:

| Parameter  | Description  |
|--|--|
| <code>credential_configuration_ids</code>                                | An array of strings, where each item is a type of credential that can be obtained from the credential issuer.  |
| <code>grants</code>  | An object that indicates to GOV.UK Wallet the grant types that the credential issuer's authorisation server is prepared to process for this credential offer. Currently, the only supported grant type is <code>grants.urn:ietf:params:oauth:grant-type:pre-authorized_code</code> . |
| <code>grants.urn:ietf:params:oauth:grant-type:pre-authorized_code</code> | The grant type required for the pre-authorised code flow.  |
| <code>grants.urn:ietf:params:oauth:grant-</code>                         | The pre-authorised code generated and signed by the credential issuer and which gives GOV.UK Wallet authorisation  |

`type:pre-authorized_code.pr` to obtain an access token from the authorisation server.

`credential_issuer` The URL of the credential issuer. This is used later by GOV.UK Wallet to fetch the credential.

This is an example of a credential offer JSON object. The `pre-authorized_code` in the example [is decoded below it](#).

```
{  
  "credential_configuration_ids": [  
    "FishingLicenceCredential"  
,  
  "grants": {  
    "urn:ietf:params:oauth:grant-type:pre-authorized_code": {  
      "pre-authorized_code": "eyJraWQiOiI1ZGNiZWU4NjNiNWQ3Y2MzMGM5YmExZjczoT  
    }  
  },  
  "credential_issuer": "https://example-credential-issuer.gov.uk"  
}
```

## The pre-authorised code

When the wallet requests a credential, GOV.UK Wallet offers them an access token signed by GOV.UK One Login. The credential issuer can validate that token to confirm the request came from a genuine GOV.UK Wallet instance, and get assured confirmation of the logged in user's walletSubjectId.

One of the methods defined in the OID4VCI specification for issuing access tokens is the [pre-authorized code flow \(\[https://openid.net/specs/openid-4-verifiable-credential-issuance-1\\\_0.html#section-3.5\]\(https://openid.net/specs/openid-4-verifiable-credential-issuance-1\_0.html#section-3.5\)\)](#). This is the only method accepted by GOV.UK Wallet.

The pre-authorised code flow makes getting credentials simpler by letting the issuer start the authorisation flow. In a standard authorisation flow, GOV.UK Wallet would need to start the flow. In the pre-authorised code flow, the credential issuer starts the authorisation flow and passes the details to GOV.UK Wallet (via the pre-authorised code) so that GOV.UK Wallet can continue.

The pre-authorised code is a JWT generated and signed by your credential issuer and included in the credential offer.

## JWT Header

The JWT header must contain the following parameters:

```
{  
  "kid": "5dcbee863b5d7cc30c9ba1f7393dacc6c16610782e4b6a191f94a7e8b1e1510f",  
  "typ": "JWT",  
  "alg": "ES256"  
}
```

- **kid** matches a **kid** in the JSON Web Key Set (JWKS) published to your [`/.well-known/jwks.json`](#) [`\(/credential-issuer-functionality/jwks\)`](#) endpoint
- **typ** must be “JWT” - this is the media type of the complete JWT
- **alg** must be “ES256” - this is the algorithm used to sign the JWT

## JWT Payload

The JWT payload must contain the following claims:

```
{  
  "clientId": "<YOUR ONE LOGIN CLIENT ID>",  
  "credential_identifiers": [  
    "<CREDENTIAL IDENTIFIER>"  
,  

```

- **clientId** is your client ID which you received when you registered your service to use GOV.UK One Login
- **credential\_identifiers** is the unique identifier for the specific credential offer - currently, GOV.UK Wallet only supports one identifier per pre-authorised code. This should be a long random value (for example, a UUIDv4) and should not be a personal identifier or account identifier.
- **exp** is the expiration date of the pre-authorised code - we recommend this is aligned with the length of time a user can remain inactive on your service, up to a maximum of 1

hour. Must be expressed in epoch time as per the [IETF RFC 7519](https://datatracker.ietf.org/doc/html/rfc7519) (<https://datatracker.ietf.org/doc/html/rfc7519>)

- **iat** is the time at which the pre-authorized code was issued. Must be expressed in epoch time as per the [IETF RFC 7519](https://datatracker.ietf.org/doc/html/rfc7519) (<https://datatracker.ietf.org/doc/html/rfc7519>)
- **iss** is the URL of your credential issuer
- **aud** is the URL of the authorisation server your credential issuer relies on for authorisation. This must be set to the GOV.UK One Login authorisation server: <https://token.account.gov.uk/>

## Signature

The pre-authorised code must be signed with an Elliptic Curve Digital Signature Algorithm (ECDSA) private key for signing. The corresponding public key, which forms a pair with the private key used for signing, must be made publicly accessible at your [/.well-known/jwks.json](#) ([/credential-issuer-functionality/jwks](#)) endpoint. This is because GOV.UK One Login will need access to the public key to verify the signature on the pre-authorised code.

The signing algorithm must be ECDSA with the P-256 curve and the SHA-256 hash function.

## Storing the credential information

Your credential issuer will need to store some context to track the credential issuance process.

You must store the:

- credential offer identifier: this is the **credential\_identifiers** in the pre-authorised code payload described above
- GOV.UK Wallet subject identifier (`walletSubjectId`): this was included in the user information your service received when the user authenticated with GOV.UK One Login
- credential details: this could be an identifier for the record to be issued as a credential and/or other relevant information about the credential itself

The credential issuer may also want to store the:

- offer creation timestamp: records when the credential offer was created
- offer expiration time: indicates when the credential offer will expire

Storing this information allows the credential issuer to track which credential is being offered and to which recipient and GOV.UK Wallet instance, and the specific context of the issuance request.

The reference implementation of the credential issuer uses AWS DynamoDB as the caching solution.

## The credential offer URL

Your credential issuer must pass the credential offer to the GOV.UK Wallet **by value** with a URL.

The URL for passing a credential offer by value follows the following format:

1. The GOV.UK Wallet “credential offer endpoint”
2. A query parameter `credential_offer` that contains the Base64Url-encoded credential offer object

For example:

```
https://mobile.account.gov.uk/wallet/add?credential_offer=%7B%22credential_c
```

## Displaying the credential offer URL

To provide a consistent experience across devices, your service’s webpage should present the credential offer URL to the user in 2 ways:

- as a QR code: this is best for users accessing your service on a separate device. They can quickly scan the code to communicate the credential offer to their GOV.UK Wallet
- as a call-to-action (CTA) link: this is best for users already on their mobile device - tapping the CTA link directly opens the credential offer in their GOV.UK Wallet

This approach provides a user-friendly way to deliver the credential offer to GOV.UK Wallet, regardless of which device the user is on.

## GOV.UK Wallet credential offer endpoints

The GOV.UK Wallet credential offer endpoint varies depending on the environment:

### Environment   Credential offer endpoint URL

---

|             |  |
|-------------|--|
| integration | <a href="https://mobile.integration.account.gov.uk/wallet/add?credential_offer=">https://mobile.integration.account.gov.uk/wallet/add?<br/>credential_offer=</a> |
|-------------|--|

---

|            |  |
|------------|--|
| production | <a href="https://mobile.account.gov.uk/wallet/add?credential_offer=">https://mobile.account.gov.uk/wallet/add?<br/>credential_offer=</a> |
|------------|--|

This page was set to be reviewed before 24 September 2025. This might mean the content is out of date.



## Accessibility

### **OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Credential

The credential issuer credential endpoint is a required endpoint defined in the [OID4VCI](#) ([https://openid.net/specs/openid-4-verifiable-credential-issuance-1\\_0.html#section-8](https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html#section-8)) specification. It's where GOV.UK Wallet, acting on behalf of the holder, requests and receives verifiable credentials from the credential issuer. Government departments acting as credential issuers must implement this endpoint according to this specification to correctly integrate with GOV.UK Wallet.

## Technical details

### Endpoint URI

The credential endpoint's URI path is implementation-specific.

The credential issuer must publish the location of their credential endpoint in their [issuer metadata API](#) (</credential-issuer-functionality/metadata/api.html#well-known-openid-credential-issuer>) using the `credential_endpoint` parameter.

### Request format

The credential endpoint must accept HTTP POST requests.

GOV.UK Wallet will send a request to the credential issuer's credential endpoint to get a verifiable credential. The credential request must include:

- the Authorization header: a bearer access token (JWT) issued by GOV.UK One Login
  - this token authorises the credential issuance
- the request body (JSON): a proof of possession token (JWT) issued by GOV.UK Wallet
  - this proves the wallet controls the private key to which the credential will be bound

### Request validation

#### Authorization header

The access token in the credential request is used to authorise the issuance of a verifiable credential. It's different from the access token used when the user initially logs in to

GOV.UK One Login, which is for authentication purposes. Because they have different roles, they are signed and verified using different keys.

The credential issuer must validate the access token to make sure that it was issued by GOV.UK One Login and that the request originates from the expected user.

To validate the access token, you should complete the following steps.

**1. Verify the signature:**

- retrieve the GOV.UK One Login JSON Web Key Set (JWKS) from its published JWKS endpoint
- extract the `kid` (key ID) parameter from the header
- find the matching public key in the JWKS by comparing `kid` values
- confirm the `alg` (algorithm) parameter in the token header matches the algorithm of the identified public key
- use the matching public key to cryptographically verify the token signature using the specified algorithm

**2. Validate the header parameters by ensuring that:**

- the value of the `typ` (type) parameter is "`at+jwt`"

Below is an example of an access token header:

```
{  
  "alg": "ES256",  
  "typ": "at+jwt",  
  "kid": "8f9ec544-f5df-4d37-a32d-a5defd78ab0f"  
}
```

**3. Validate the payload claims by ensuring that:**

- the value of the `iss` (issuer) claim matches the GOV.UK One Login URL:  
`"https://token.integration.account.gov.uk"` (integration) or  
`"https://token.account.gov.uk"` (production)
- the value of the `aud` (audience) claim is the credential issuer URL
- the value of the `sub` (subject - this is the wallet subject identifier) claim matches the value stored in your cache for this specific credential issuance flow
- the value of the `exp` (expiration time) claim is in the future
- the value of the `credential_identifiers` claim matches the value stored in your cache for this specific credential issuance flow

- the value of the `c_nonce` claim matches the value of the `nonce` claim in the [proof of possession \(/credential-issuer-functionality/credential/#request-body\)](#)
- this API has not received another access token with the same `jti` (JWT ID) that is still within its validity period

Below is an example of an access token payload:

```
{
  "sub": "urn:fdc:wallet.account.gov.uk:2024:DtPT8x-dp_73tnLY3KNTiCitziN9GEh",
  "iss": "https://token.account.gov.uk",
  "aud": "https://example-credential-issuer.gov.uk",
  "exp": 1756115975,
  "credential_identifiers": [
    "daa01d3e-b17c-4c8a-8adf-ef808b456c9c"
  ],
  "c_nonce": "657a09cd-7165-486d-a858-065eb23f7a8d",
  "jti": "62b45850-4c5c-4696-983a-af66450301d4"
}
```

The `sub` claim validation is an important security control. This pairwise identifier, which starts with `urn:fdc:wallet.account.gov.uk:`, is generated by GOV.UK One Login for each user's wallet instance and helps ensure the credential is issued to the right user.

Your implementation must compare the `sub` value from the access token against the wallet subject identifier you got and stored when the user authenticated with GOV.UK One Login. This comparison makes sure that the wallet requesting the credential belongs to the same user who authenticated with your service.

 **If the identifiers do not match, the wallet trying to get the credential does not belong to the person logged in to your service. Your credential issuer must stop the issuance flow and consider logging the attempt for audit and fraud prevention.**

## Request body

The request body sent to the credential endpoint must be a JSON object containing proof of possession that demonstrates the wallet's control of the private key to which the credential will be bound.

It must contain the following parameters:

| Parameter        | Description  | Value(s)      |
|------------------|--|---------------|
| proof            | A JSON containing the proof of possession of the cryptographic key material. |               |
| proof.proof_type | A string indicating the type of proof being presented.                       | Must be jwt . |
| proof.jwt        | The JSON Web Token (JWT) that serves as the proof.                           |               |

Below is an example of a request body:

```
{
  "proof": {
    "proof_type": "jwt",
    "jwt": "ew0KICAiYWxnIjogIkVTMjU2IiwNCiAgInR5cCI6ICJvcGVuaWQ0dmNpLXByb29
  }
}
```

The proof of possession is a cryptographic mechanism that verifies the wallet controls the private key that matches the public key (“did:key”) that will be associated with the credential. This ensures credentials are issued to their rightful holder.

This token is generated by GOV.UK Wallet and includes a cryptographic client `nonce` (from the access token issued by GOV.UK One Login) that has been signed with the wallet’s private signing key.. The `did:key` (the wallet’s public key) is included in the token’s header `kid` parameter.

When your credential issuer receives the credential request, it verifies the proof of possession signature with the `did:key` . Successful verification shows the wallet’s ownership of the private key corresponding to that public `did:key` .

There is more information about [the did:key method \(/credential-issuer-functionality/credential/#the-did-key-format\)](#).

To validate the proof, you should complete the following steps.

#### 1. Verify the signature:

- extract the `kid` (key ID) parameter from the header, which contains the wallet’s `did:key`
- convert the `did:key` value to its corresponding public key

- confirm the `alg` (algorithm) parameter in the proof header is `ES256` and is compatible with the key type derived from the `did:key`
- use the public key to cryptographically verify the proof signature using the specified algorithm

## 2. Validate the header parameters by ensuring that:

- the value of the `typ` (type) parameter is "`openid4vci-proof+jwt`"

Below is an example of a proof header:

```
{
  "alg": "ES256",
  "typ": "openid4vci-proof+jwt",
  "kid": "did:key:zDnaeSGFSQMYvnLbLWEubhhGDPoq7pA9MMNvumvbsmMCZovUR"
}
```

## 3. Validate the payload claims by ensuring that:

- the value of the `iss` (issuer) claim matches the GOV.UK Wallet identifier - `urn:fdc:gov:uk:wallet`
- the value of the `aud` (audience) claim is the credential issuer URL
- the value of the `iat` (issued at) is a numeric date formatted as seconds since the epoch - this must be a date in the past that is after the pre-authorized code was generated
- the value of the `nonce` claim matches the value of the `c_nonce` claim in the access token

Below is an example of a proof of possession token payload:

```
{
  "iss": "urn:fdc:gov:uk:wallet",
  "aud": "https://example-credential-issuer.gov.uk",
  "iat": 1745233623816,
  "nonce": "bd423745-7705-45c2-9f51-6ae8dcac5589"
}
```

More information about the credential request can be found in the [OID4VCI specification](https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html#section-8.2) ([https://openid.net/specs/openid-4-verifiable-credential-issuance-1\\_0.html#section-8.2](https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html#section-8.2)).

# Successful response format

After validating the request successfully, the credential endpoint must return a 200 OK HTTP status code and a JSON response following the [OID4VCI specification](https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html#section-8.3) ([https://openid.net/specs/openid-4-verifiable-credential-issuance-1\\_0.html#section-8.3](https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html#section-8.3)). The response contains the issued credential as a JWT and a unique notification ID (a generated UUIDv4) for the callback success/failure notification:

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store

{
  "credentials": [
    {
      "credential": "eyJraWQiOiJkaWQ6d2Vi0mV4YW1wbGUtY3JlZGVudGlhbC1pc3N1ZXI",
    },
    "notification_id": "776aefd4-26c6-4a5f-aa7c-b5e294cd87cd"
  ]
}
```

The `notification_id` should only be included in the response if the credential issuer implements the [notification endpoint \(/credential-issuer-functionality/notification\)](#).

More information about the credential response can be found in the [OID4VCI specification](https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html#section-8.3) ([https://openid.net/specs/openid-4-verifiable-credential-issuance-1\\_0.html#section-8.3](https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html#section-8.3)).

The steps for constructing and issuing the verifiable credential are:

1. Retrieve the underlying data that will go into the credential from your database using the information in your issuance cache.
2. Build the verifiable credential according to the [W3C Verifiable Credentials Data Model v2.0](https://www.w3.org/TR/vc-data-model-2.0/) (<https://www.w3.org/TR/vc-data-model-2.0/>).
3. [Cryptographically bind the credential to the wallet \(/credential-issuer-functionality/credential/#further-guidance-on-credential-binding\)](#) by using the wallet's `did:key` as the credential's subject identifier (the `sub` claim).
4. Sign the credential with your issuer's private key.

The example below shows the structure of a verifiable credential using a JSON Web Token (JWT) format for a fishing licence.

## Header

```
{
  "alg": "ES256",
  "typ": "vc+jwt",
  "cty": "vc",
  "kid": "did:web:example-credential-issuer.gov.uk#5dcbee863b5d7cc30c9ba1f73"
}
```

- **alg** (algorithm). REQUIRED. The cryptographic algorithm used to sign the JWT - must be "ES256" for ECDSA using the P-256 curve.
- **typ** (type). REQUIRED. The media type of the signed content - must be "vc+jwt".
- **cty** (content type). REQUIRED. The media type of the secured content (the payload) - must be "vc".
- **kid** (key ID). REQUIRED. The DID URL of the issuer's public key used for signature verification - must match the **id** of the corresponding key in the credential issuer's DID Document, to let recipients locate the correct public key for signature verification.

## Payload

```
{
  "iss": "https://example-credential-issuer.gov.uk",
  "sub": "did:key:ebfaeb1fd712ebf1c276e12ec21",
  "iat": "1712664731",
  "@context": [
    "https://www.w3.org/ns/credentials/v2",
    "<JSON-LD CONTEXT URI FOR ISSUER>"
  ],
  "type": [
    "VerifiableCredential",
    "FishingLicenceCredential"
  ],
  "issuer": "https://example-credential-issuer.gov.uk",
  "name": "Fishing licence",
  "description": "Permit for fishing activities",
  "validFrom": "2024-04-09T12:12:11Z",
  "validUntil": "2028-12-10T22:59:59Z",
  "credentialSubject": {
    "id": "did:key:ebfaeb1fd712ebf1c276e12ec21",
    "name": [
      {
        "name": "John Doe",
        "value": "John Doe"
      }
    ]
  }
}
```

```

    "nameParts": [
        {
            "value": "Sarah",
            "type": "GivenName"
        },
        {
            "value": "Edwards",
            "type": "FamilyName"
        }
    ],
    "fishingLicenceRecord": [
        {
            "licenceNumber": "009878863",
            "issuanceDate": "2023-12-10",
            "expiryDate": "2028-12-10"
        }
    ]
}

```

- **iss** (issuer). REQUIRED. The URL of the credential issuer service operated by the organisation sharing the credential.
- **sub** (subject). REQUIRED. The identifier of the holder of the information in the credential. The subject identifier is a decentralised identifier **did:key** generated by the wallet. In the credential issuance flow, the wallet shares its **did:key** with the issuer, and the issuer makes this the value of the credential's **sub** claim. This cryptographically binds the credential to the wallet.
- **iat** (issued at). OPTIONAL. The time at which the JWT was issued. Must be expressed in epoch time as per the [IETF RFC 7519](https://datatracker.ietf.org/doc/html/rfc7519) (<https://datatracker.ietf.org/doc/html/rfc7519>).
- **@context** . REQUIRED. The context of the data exchange. It must be a set of URIs that point to documents that describe the context. The first item in the set must be the URI "<https://www.w3.org/ns/credentials/v2>" .
- **type** . REQUIRED. A set of values indicating the type of verifiable credentials issued by the issuer. The first value in the set must be **VerifiableCredential**
- **issuer** . REQUIRED. The URL of the credential issuer service operated by the organisation sharing the credential. Must be the same as the value of the **iss** claim.

- `name` . OPTIONAL. Issuer-specified credential name.
- `description` . OPTIONAL. Issuer-specified credential description.
- `validFrom` . OPTIONAL. It represents the date and time the credential becomes valid. Must be expressed in ISO 8601 format with seconds ( `YYYY-MM-DDTHH:mm:ssZ` ) as per the [VC data model v2.0](https://www.w3.org/TR/vc-data-model-2.0/) (<https://www.w3.org/TR/vc-data-model-2.0/>).
- `validUntil` . REQUIRED. It represents the date and time the credential stops being valid. Must be expressed in ISO 8601 format with seconds ( `YYYY-MM-DDTHH:mm:ssZ` ) as per the [VC data model v2.0](https://www.w3.org/TR/vc-data-model-2.0/) (<https://www.w3.org/TR/vc-data-model-2.0/>).
- `credentialSubject` . REQUIRED. An object containing claims about the holder of the verifiable credential.

### Guidance on credential expiration

The `validUntil` claim in the credential specifies the date after which any consumer of the verifiable credential must consider it invalid (expired) and reject it. The entitlement in the underlying data source (the fishing license record in the example) may still be valid, as represented by the `expiryDate` but this digital credential representation of the underlying representation has expired. If the `validUntil` date has passed, the holder must get a new verifiable credential to update the digital representation of their entitlement.

Credentials issued to GOV.UK Wallet must include the `validUntil` claim. This date must be set to the same date as, or earlier than, the `credentialSubject.expiryDate`.

Always consider the expiration of the underlying credential when setting JWT expiration.

### Guidance on including photos

If a photo is required in the credential, you must include it within the `credentialSubject` claim as a Base64-encoded string.

GOV.UK Wallet will validate the image to make sure that:

- The image's format is JPG or PNG conforming to one of the following specifications:
  - `FF D8 FF E0 JPG`
  - `FF D8 FF EE JPG`
  - `FF D8 FF DB JPG`
  - `89 50 4E 47 0D 0A 1A 0A PNG`
  - `FF D8 FF E0 00 10 4A 46 49 46 00 01 JFIF`
- The image must not exceed 1 MiB (1 Mebibyte = 1,048,576 bytes) in size before encoding to Base64
- The image must have EXIF metadata stripped

If the GOV.UK Wallet fails to process an image in a credential, a [credential\\_failure](#) error will be returned, following the [notification specification \(/credential-issuer-functionality/notification\)](#).

Below are examples of claims representing PNG and JPEG images respectively, encoded in Base64 format:

```
"photo": "iVBORw0KGgoAAAANSUhIAAAAE2BViAAAA[...]" // PNG file as source
```

```
"photo": "/9j/4AAQSkZJRgABAQpDYXQwMy5qcGf/[...]" // JPG file as source
```

## Signature

The credential must be signed with your credential issuer's private signing key using the ECDSA (Elliptic Curve Digital Signature Algorithm) cryptographic algorithm with P-256 (also known as Secp256r1) elliptic curve.

## Error response format

If the credential request could not be processed successfully, the credential issuer must return an appropriate HTTP error status code.

When the credential request does not include an access token or the access token is invalid, the credential endpoint must return a 401 Unauthorized HTTP status code and include the [WWW-Authenticate](#) response header field as defined in [RFC6750](#) (<https://datatracker.ietf.org/doc/html/rfc6750#section-3>), specifying the [Bearer](#) authentication scheme.

This is an example error response to a request with no access token:

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Bearer
Cache-Control: no-store
```

This is an example error response to an authentication attempt using an invalid access token:

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Bearer error="invalid_token"
Cache-Control: no-store
```

When there is an issue with the request body, the credential endpoint must return a 400 Bad Request HTTP status code and a response body in JSON format containing the following parameter:

- `error` : A case-sensitive string indicating the error - `invalid_proof` (the proof of possession is invalid), or `invalid_nonce` (the `nonce` is invalid or does not match the access token's `c_nonce` )

Below is an example of a credential error response when the credential request included an invalid proof of possession:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
Cache-Control: no-store

{
  "error": "invalid_proof"
}
```

## Further guidance on credential binding

### Binding credentials to users

Because each GOV.UK Wallet instance can be uniquely identified, your service can bind a credential with a specific wallet instance. GOV.UK Wallet uses a specific type of [decentralised identifier \(DID\) \(https://www.w3.org/TR/did-core/\)](https://www.w3.org/TR/did-core/) called a `did:key` to cryptographically bind credentials to a user's wallet.

A [did:key \(https://w3c-ccg.github.io/did-method-key/\)](https://w3c-ccg.github.io/did-method-key/) is a DID method. The DID represents the public key of an asymmetric key pair generated when GOV.UK Wallet is installed on a device. The private key never leaves the device, whereas the `did:key` is shared with credential issuers and verifiers. This allows credentials to be cryptographically bound to a specific GOV.UK Wallet instance.

The GOV.UK Wallet creates a `did:key` from a **P-256** (also known as Secp256r1) elliptic curve public key.

### The `did:key` format

The `did:key` method is used to transfer public keys.

The format of a `did:key` is `did:key:multibaseValue`. The `multibaseValue` is the base58-btc multibase string representation of concatenating the multicodec identifier for the public key type and the compressed public key.

```
did-key-format := did:key:MULTIBASE(base58-btc, MULTICODEC(public-key-type,
```

In Elliptic Curve Cryptography (ECC), the public key is a pair of **x** and **y** coordinates. A compressed public key is the **x** coordinate, which is 32 bytes in length, with a prefix, of 1 byte in length, that indicates whether the **y** coordinate is even or odd. The prefix is **02** if the **y** coordinate is even and **03** if it is odd. The resulting compressed public key is **33 bytes** in length:

```
Public key: 52972572d465d016d4c501887b8df303eee3ed602c056b1eb09260dfa0da0ab2
```

```
Public key (x coordinate): 52972572d465d016d4c501887b8df303eee3ed602c056b1eb
```

```
Public key (y coordinate): 88742f4dc97d9edb6fd946bab002fdfb06f26caf117b9405
```

```
// y coordinate is even so "02" is prepended to the x coordinate
```

```
Public key (compressed): 0252972572d465d016d4c501887b8df303eee3ed602c056b1eb
```

The **multibaseValue** is generated as follows:

1. Encode the compressed public key as bytes
2. Prefix the key bytes with the **curve multicodec value** encoded as **unsigned varint** (variable length integers)
3. the multicodec hexadecimal value of a P-256 elliptic curve public key is **0x1200**, in varint-encoded bytes that is **[0x80, 0x24]**
4. Encode the above with **base58-btc** and then prefix it with "**z**" to indicate the base58-btc encoding - the result is the **multibaseValue**

The following is an example of a **did:key** derived from a base-58 encoded P-256 public key:

```
did:key:zDnaewZMz7MN6xSaAFADkDZJzMLbGSV25uKHAeXaxnPCwZomX
```

All DIDs derived from a P-256 public key always start with "**zDn**".

## Verifying a credential

To share a verifiable credential with a verifier, the wallet creates a verifiable presentation containing the verifiable credential and signs the credential with the wallet's private key.

The verifier must be able to confirm that the system presenting the credential (GOV.UK Wallet) is also the intended holder of that credential. The verifier must confirm **proof of possession** of the verifiable credential. This is done by verifying that the entity which

signed the verifiable presentation is the same as the subject of the verifiable credential. In other words, the `did:key` in the verifiable credential must be able to verify the signature on the verifiable presentation.

This page was last reviewed on 25 August 2025. It needs to be reviewed again on 25 February 2026 .



## Accessibility

## **OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

Table of contents

# API

## /.well-known/did.json

### get

A public endpoint for GOV.UK Wallet and credential verifiers to retrieve the credential issuer's public keys for verifying credentials

### Responses

#### Status Description

200 Credential issuer's DID document.

```
{
  "@context": [
    "https://www.w3.org/ns/did/v1",
    "https://w3id.org/security/suites/jws-2020/v1"
  ],
  "id": "did:web:example-credential-issuer.gov.uk",
  "verificationMethod": [
    {
      "id": "did:web:example-credential-issuer.gov.uk#5dcbee863b5d7cc30c9ba1f7393dacc6c16610782e4b6a191f94a7e8b1e1510f",
      "type": "JsonWebKey2020",
      "controller": "did:web:example-credential-issuer.gov.uk",
      "publicKeyJwk": {
        "kty": "EC",
        "kid": "5dcbee863b5d7cc30c9ba1f7393dacc6c16610782e4b6a191f94a7e8b1e1510f",
        "crv": "P-256",
        "x": "6jCKX_QRrmTeEJi-uiwcYqu8BgMg170g2pdAst24MPE",
        "y": "icPzjbSk6apD_SNvQt8NWOPlPeGG4KYU55GfnARryoY",
        "alg": "ES256"
      }
    }
  ],
  "assertionMethod": [
    "did:web:example-credential-issuer.gov.uk#5dcbee863b5d7cc30c9ba1f7393dacc6c16610782e4b6a191f94a7e8b1e1510f"
  ]
}
```

## Schemas

### DidResponse

| Name     | Type   | Required | Description   | Schema |
|----------|--------|----------|---|--------|
| @context | array  | true     | An array of URL contexts which define the terms used in the DID document. |        |
| id       | string | true     | The unique identifier of the DID document.                                |        |

| Name               | Type  | Required | Description   | Schema                             |
|--------------------|-------|----------|---|------------------------------------|
| verificationMethod | array | true     | An array of verification methods (cryptographic public keys).                                       | <a href="#">VerificationMethod</a> |
| assertionMethod    | array | true     | Array of DID URLs where each URL uniquely identifies a verification method within the DID document. |                                    |

## VerificationMethod

| Name         | Type   | Required | Description | Schema                       |
|--------------|--------|----------|-------------|------------------------------|
| id           | string | true     |             |                              |
| type         | string | true     |             |                              |
| controller   | string | true     |             |                              |
| publicKeyJwk | object | true     |             | <a href="#">PublicKeyJwk</a> |

## PublicKeyJwk

| Name | Type   | Required | Description   | Schema |
|------|--------|----------|---|--------|
| kty  | string | true     | Key Type. The family of cryptographic algorithms used with the key. |        |
| kid  | string | true     | Key ID. Unique identifier to match a specific key.                  |        |
| crv  | string | true     | Curve. Cryptographic curve used with the key.                       |        |
| x    | string | true     | The “x” coordinate for the elliptic curve point.                    |        |
| y    | string | true     | The “y” coordinate for the elliptic curve point.                    |        |
| alg  | string | true     | Algorithm. The cryptographic algorithm used with the key.           |        |

This page was set to be reviewed before 5 September 2025. This might mean the content is out of date.



## [Accessibility](#)

**OGL** All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

© Crown copyright

[Table of contents](#)

# DID document

The DID document endpoint lets GOV.UK Wallet verify the credentials it receives from credential issuers, who must use decentralised identifiers (DIDs).

This endpoint exposes the credential issuer's DID document, which contains the public cryptographic keys used to verify signatures on credentials issued by the credential issuer. This process checks the credentials' authenticity and integrity.

## Technical details

### Endpoint location

The DID document must be publicly accessible at the standardised location `/well-known/did.json` on the credential issuer's domain.

### Response format

The endpoint must return a 200 OK HTTP status code and a valid JSON response that follows the [W3C Decentralized Identifiers \(DIDs\) v1.0 specification](https://www.w3.org/TR/did-1.0/) (<https://www.w3.org/TR/did-1.0/>).

The DID document must include the following parameters:

- `@context` : A set of URI references that define the meaning of the terms used in the DID document, allowing its properties to be correctly interpreted
- `id` : The decentralised identifier of the credential issuer in the format `"did:web:<CREDENTIAL-ISSUER-DOMAIN>"`
- `verificationMethod` : An array of verification methods containing the credential issuer's public keys
- `assertionMethod` : An array listing which verification methods can be used for making assertions - in the context of the credential issuer, this means issuing credentials

Each object in the array must contain:

- **`id`** : A unique identifier for the verification method in the format "`did:web:<CREDENTIAL-ISSUER-DOMAIN>#<KEY-ID>`" - this corresponds to the `kid` (key ID) parameter in the header of issued credentials
- **`type`** : The cryptographic suite used for verification - this must be `JsonWebKey2020` as the [JSON Web Signature 2020 specification](https://www.w3.org/community/reports/credentials/CG-FINAL-lds-jws2020-20220721/) (<https://www.w3.org/community/reports/credentials/CG-FINAL-lds-jws2020-20220721/>) is used
- **`controller`** : The entity (the credential issuer in this case) that has the authority to use the private key associated with the verification method - this must be in the format "`did:web:<CREDENTIAL-ISSUER-DOMAIN>>`"
- **`publicKeyJwk`** : The credential issuer's public key - this is represented as a JSON Web Key (JWK) and contains parameters specific to the elliptic curve algorithm used (P-256)

Each object in the `verificationMethod` array represents a public key, which is the counterpart to the private key held securely by the credential issuer. These private keys are used to sign credentials. GOV.UK Wallet only accepts credentials signed using an elliptic curve key based on the P-256 curve.

Not all public keys listed in the `verificationMethod` are authorised for credential issuance. The `assertionMethod` array acts as an allow list, specifying which of those keys are trusted for credential issuance. Each item in `assertionMethod` directly references a public key in the `verificationMethod` array using its unique ID.

When GOV.UK Wallet receives a credential issued by the credential issuer, it uses the `kid` (key ID) in the credential's header to find the matching public key in the `verificationMethod` array. Then, it checks if the same verification method ID is present in the `assertionMethod` array. This process confirms that the key used to sign the credential was authorised for credential issuance by the DID controller (the credential issuer).

## DID document example

Below is an example of a DID document containing an elliptic curve key based on the P-256 curve in the verification method:

```
{
  "@context": [
    "https://www.w3.org/ns/did/v1",
    "https://w3id.org/security/suites/jws-2020/v1"
  ],
  "id": "did:web:example-credential-issuer.gov.uk",
  "verificationMethod": [
    {
      "id": "did:web:example-credential-issuer.gov.uk#key-1",
      "type": "JsonWebKey2020",
      "controller": "did:web:example-credential-issuer.gov.uk",
      "publicKeyJwk": {
        "crv": "P-256",
        "x": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6q7r8s9t0u1v2w3x4y5z6",
        "y": "b3d5e7f9g1h3i5j7k9l1m3n5o7p9q1r5s3t7u9v1w5x3z7",
        "crv": "P-256",
        "x": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6q7r8s9t0u1v2w3x4y5z6",
        "y": "b3d5e7f9g1h3i5j7k9l1m3n5o7p9q1r5s3t7u9v1w5x3z7"
      }
    }
  ]
}
```

```
{  
  "id": "did:web:example-credential-issuer.gov.uk#5dcbee863b5d7cc30c9b",  
  "type": "JsonWebKey2020",  
  "controller": "did:web:example-credential-issuer.gov.uk",  
  "publicKeyJwk": {  
    "kty": "EC",  
    "kid": "5dcbee863b5d7cc30c9ba1f7393dacc6c16610782e4b6a191f94a7e8b",  
    "crv": "P-256",  
    "x": "6jCKX_QRrmTeEJi-uiwcYqu8BgMgl70g2pdAst24MPE",  
    "y": "icPzjbSk6apD_SNvQt8NWOPlPeGG4KYU55GfnARryoY",  
    "alg": "ES256"  
  },  
},  
]  
,  
  "assertionMethod": [  
    "did:web:example-credential-issuer.gov.uk#5dcbee863b5d7cc30c9ba1f7393d",  
  ]  
}
```

This page was set to be reviewed before 3 October 2025. This might mean the content is out of date.



## Accessibility

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# API

A public endpoint for GOV.UK One Login to retrieve the credential issuer's JSON Web Key Set (JWKS) of public keys which can be used to verify the pre-authorised code.

## /.well-known/jwks.json

### get

A public endpoint that stores the JSON Web Key Set (JWKS) of public keys issued by a service. These keys can be used by client applications to verify the signature of a JSON Web Token (JWT).

#### Responses

| Status | Description               | Schema                       |
|--------|---------------------------|------------------------------|
| 200    | Credential issuer's JWKS. | <a href="#">JwksResponse</a> |

```
{  
  "keys": [  
    {  
      "kty": "EC",  
      "use": "sig",  
      "crv": "P-256",  
      "kid": "5dcbee863b5d7cc30c9ba1f7393dacc6c16610782e4b6a191f94a7e8b1e1510f",  
      "x": "6jCKX_QRrmTeEJi-uiwcYqu8BgMg170g2pdAst24MPE=",  
      "y": "icPzjbSk6apD_SNvQt8NW0P1PeGG4KYU55GfnARryoY=",  
      "alg": "ES256"  
    }  
  ]  
}
```

# Schemas

## JwksResponse

| Name | Type  | Required | Description  | Schema              |
|------|-------|----------|--|---------------------|
| keys | array | true     | A set of public keys, each in JSON Web Key (JWK) format. | <a href="#">Key</a> |

## Key

| Name | Type   | Required | Description   | Schema |
|------|--------|----------|---|--------|
| kty  | string | true     | Key Type. The family of cryptographic algorithms used with the key. |        |
| kid  | string | true     | Key ID. Unique identifier to match a specific key.                  |        |
| crv  | string | true     | Curve. Cryptographic curve used with the key.                       |        |
| x    | string | true     | The “x” coordinate for the elliptic curve point.                    |        |
| y    | string | true     | The “y” coordinate for the elliptic curve point.                    |        |
| alg  | string | true     | Algorithm. The cryptographic algorithm used with the key.           |        |
| use  | string | true     | The intended use of the key.  |        |

This page was set to be reviewed before 5 September 2025. This might mean the content is out of date.



## Accessibility

## **OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# JSON Web Key Set (JWKS)

The JWKS endpoint is a required endpoint that exposes the credential issuer's public cryptographic keys, which can be used by GOV.UK Wallet to verify the pre-authorised code, in JSON Web Key Set (JWKS) format.

This endpoint lets the GOV.UK One Login retrieve the credential issuer's public keys and then verify the [pre-authorised code \(/credential-issuer-functionality/credential-offer/#the-pre-authorised-code\)](#) signature. This process confirms that the pre-authorised code was issued by the expected credential issuer and that it has not been tampered with.

## Technical details

### Endpoint location

The JWKS must be publicly accessible at the standardised location [/.well-known/jwks.json](#) on the credential issuer's domain.

### Response format

The endpoint must return a 200 OK HTTP status code and a valid JSON response that follows the JWKS specification defined in [RFC 7517](#) (<https://datatracker.ietf.org/doc/html/rfc7517>). Each key within the JWKS is represented as a JSON Web Key (JWK) object. The JWKS usually contains only one key, but it can contain two keys during a key rotation overlap period.

The JWK for an elliptic curve key based on the P-256 curve must include the following parameters:

- **kty** : The family of cryptographic algorithms used with the key - must be “EC”.
- **kid** : A unique identifier for a specific key within the set - this value will be referenced in the pre-authorised code header to show which key was used for signing and which key must be used for verification. This parameter is important for associating the correct public key with the pre-authorised code being verified.
- **crv** : Cryptographic curve used with the key - must be “P-256”.
- **x** : The “x” coordinate for the elliptic curve point.
- **y** : The “y” coordinate for the elliptic curve point.

- **alg** : The cryptographic algorithm used with the key - must be “ES256”.
- **use** : The intended use of the key - must be “sig” to indicate a signing key.

## JWKS example

Below is an example of a JWKS containing one elliptic curve key based on the P-256 curve:

```
{  
  "keys": [  
    {  
      "kty": "EC",  
      "use": "sig",  
      "crv": "P-256",  
      "kid": "5dcbee863b5d7cc30c9ba1f7393dacc6c16610782e4b6a191f94a7e8b1e151",  
      "x": "6jCKX_QRrmTeEJi-uiwcYqu8BgMgl70g2pdAst24MPE",  
      "y": "icPzjbSk6apD_SNvQt8NW0PlPeGG4KYU55GfnARryoY",  
      "alg": "ES256"  
    }  
  ]  
}
```

This page was set to be reviewed before 3 October 2025. This might mean the content is out of date.



## Accessibility

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# API

## /.well-known/openid-credential-issuer

### get

A public endpoint for the GOV.UK Wallet to retrieve metadata about the credential issuer.

#### Responses

| Status | Description                   | Schema                           |
|--------|-------------------------------|----------------------------------|
| 200    | Credential issuer's metadata. | <a href="#">MetadataResponse</a> |

```
{  
  "credential_issuer": "https://example-credential-issuer.gov.uk",  
  "authorization_servers": [  
    "https://token.account.gov.uk"  
,  
  "credential_endpoint": "https://example-credential-issuer.gov.uk/credential",  
  "notification_endpoint": "https://example-credential-issuer.gov.uk/notification",  
  "credential_configurations_supported": {  
    "FishingLicenceCredential": {  
      "format": "jwt_vc_json",  
      "credential_definition": {  
        "type": [  
          "VerifiableCredential",  
          "FishingLicenceCredential"  
        ]  
      },  
      "cryptographic_binding_methods_supported": [  
        "did"  
      ],  
      "credential_signing_alg_values_supported": [  
        "ES256"  
      ],  
      "proof_types_supported": {  
        "jwt": {  
          "proof_signing_alg_values_supported": [  
            "ES256"  
          ],  
          "key_attestations_required": null  
        }  
      },  
      "display": [  
        {  
          "name": "Fishing Licence number",  
          "locale": "en-GB",  
          "background_color": "#12107c",  
          "text_color": "#FFFFFF"  
        },  
        {  
          "name": "Rhif Trwydded Pysgota",  
          "locale": "en-CY",  
        }  
      ]  
    }  
  }  
}
```

| Status | Description  | Schema |
|--------|--|--------|
|        | <pre>         "background_color": "#12107c",         "text_color": "#FFFFFF"     }, ], "credentialSubject": [     "name": [         {             "nameParts": [                 {                     "display": [                         {                             "name": "Name",                             "locale": "en-GB"                         },                         {                             "name": "Enw",                             "locale": "cy-GB"                         }                     ]                 }             ]         }     ],     "fishingLicenceRecord": {         "licenceNumber": {             "display": [                 {                     "name": "Fishing Licence number",                     "locale": "en-GB"                 },                 {                     "name": "Rhif Trwydded Pysgota",                     "locale": "cy-GB"                 }             ]         },         "expirationDate": {             "display": [                 {                     "name": "Dyddiad dod i ben",                     "locale": "en-GB"                 },                 {                     "name": "Rhif Trwydded Pysgota",                     "locale": "cy-GB"                 }             ]         }     } } ] } </pre> |        |

## Schemas

### MetadataResponse

| Name                                | Type   | Required | Description  | Schema  |
|-------------------------------------|--------|----------|--|---|
| credential_issuer                   | string | true     | URL of the credential issuer.  |   |
| credential_endpoint                 | string | true     | URL of the credential issuer's credential endpoint.  |   |
| notification_endpoint               | string | false    | URL of the credential issuer's notification endpoint.  |   |
| authorization_servers               | array  | true     | Set containing the URL of the authorization server(s) the credential issuer relies on for authorization. |   |
| credential_configurations_supported | object | true     | Information about the credential(s) issued by the credential issuer.                                     | <a href="#">CredentialConfigurationsSupported</a> |

## CredentialConfigurationsSupported

Information about the credential(s) issued by the credential issuer.

| Name           | Type   | Required | Description                                   | Schema                         |
|----------------|--------|----------|---|--------------------------------|
| fishingLicence | object | true     | A credential issued by the credential issuer. | <a href="#">FishingLicence</a> |

## FishingLicence

A credential issued by the credential issuer.

| Name                                    | Type   | Required | Description  | Schema                               |
|---|--------|----------|--|--------------------------------------|
| format                                  | string | true     | Format of the credential.                                  |                                      |
| credential_definition                   | object | true     | Description of the credential type.                        | <a href="#">CredentialDefinition</a> |
| cryptographic_binding_methods_supported | array  | true     | Set of methods available for cryptographically binding the |                                      |

| Name                                    | Type   | Required | Description   | Schema                              |
|---|--------|----------|---|-------------------------------------|
|   |        |          | issued credential.  |                                     |
| credential_signing_alg_values_supported | array  | true     | Set of algorithms that the credential issuer uses to sign the credential. |                                     |
| proof_types_supported                   | object | true     | Key proof(s) supported by the credential issuer.                          | <a href="#">ProofTypesSupported</a> |

## CredentialDefinition

Description of the credential type.

| Name | Type  | Required | Description | Schema |
|------|-------|----------|-------------|--------|
| type | array | true     |             |        |

## ProofTypesSupported

Key proof(s) supported by the credential issuer.

| Name | Type   | Required | Description | Schema              |
|------|--------|----------|-------------|---------------------|
| jwt  | object | true     |             | <a href="#">Jwt</a> |

## Jwt

| Name                               | Type  | Required | Description | Schema |
|------------------------------------|-------|----------|-------------|--------|
| proof_signing_alg_values_supported | array | true     |             |        |

This page was last reviewed on 9 June 2025. It needs to be reviewed again on 9 December 2025 .



## [Accessibility](#)

**OGL** All content is available under [Open Government Licence v3.0](#), except where otherwise stated

© Crown copyright

[Table of contents](#)

# Metadata

The metadata endpoint is a required endpoint that provides essential configuration information about the credential issuer.

This endpoint lets GOV.UK Wallet and verifiers dynamically learn information about the credential issuer, such as:

- the endpoints used in the issuance flow
- the supported credential types
- how credentials should be displayed in the wallet

## Technical details

### Endpoint location

The metadata must be publicly accessible at the standardised location `/well-known/openid-credential-issuer` on the credential issuer's domain. The data published is non-sensitive metadata about the service.

### Response format

The endpoint must return a 200 OK HTTP status code and valid JSON response that follows the [OID4VCI specification](https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html#name-metadata) ([https://openid.net/specs/openid-4-verifiable-credential-issuance-1\\_0.html#name-metadata](https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html#name-metadata)).

The metadata must include the following parameters:

- `credential_issuer` : The URL of the credential issuer.
- `authorization_servers` : An array of URLs for the authorisation servers the credential issuer relies on for authorisation. This must be set to the GOV.UK One Login URL.
- `credential_endpoint` : The URL of the credential issuer's [credential endpoint](#) ([/credential-issuer-functionality/credential/#credential](#)), where credentials can be obtained.
- `credential_configurations_supported` : An object describing the credentials offered by the credential issuer.

If your credential issuer implements the optional [notification endpoint \(/credential-issuer-functionality/notification/#notification\)](#), then the metadata must include the `notification_endpoint` parameter.

You can define and use additional metadata parameters.

## Credential information

The `credential_configurations_supported` object contains key/value pairs, where each key is a unique identifier of a verifiable credential supported by the credential issuer and the value is the configuration of that verifiable credential.

Each credential object in `credential_configurations_supported` must include the following parameters:

- `format`
- `credential_definition`
- `cryptographic_binding_methods_supported`
- `credential_signing_alg_values_supported`
- `proof_types_supported`

There is more information about [the `credential\_configurations\_supported` parameter](#) ([https://openid.net/specs/openid-4-verifiable-credential-issuance-1\\_0.html#section-11.2.4](https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html#section-11.2.4)).

## Metadata example

Below is an example of a credential issuer metadata:

```
{  
  "credential_issuer": "https://example-credential-issuer.gov.uk",  
  "authorization_servers": ["https://token.account.gov.uk"],  
  "credential_endpoint": "https://example-credential-issuer.gov.uk/credentials",  
  "notification_endpoint": "https://example-credential-issuer.gov.uk/notifications",  
  "credential_configurations_supported": {  
    "FishingLicenceCredential": {  
      "format": "jwt_vc_json",  
      "credential_definition": {  
        "type": [  
          "VerifiableCredential",  
          "FishingLicenceCredential"  
        ]  
      },  
      "key": "FishingLicenceCredential"  
    }  
  }  
}
```

```
"cryptographic_binding_methods_supported": [  
    "did"  
,  
    "credential_signing_alg_values_supported": [  
        "ES256"  
,  
        "proof_types_supported": {  
            "jwt": {  
                "proof_signing_alg_values_supported": [  
                    "ES256"  
                ]  
            }  
        }  
    }  
}
```

This page was set to be reviewed before 3 October 2025. This might mean the content is out of date.



## Accessibility

## **OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# API

## /notification

### post

An endpoint used by GOV.UK Wallet to notify the credential issuer of events concerning issued credentials.

#### Responses

| Status | Description  | Schema  |
|--------|--------------|---|
| 204    | No Content   |   |
| 400    | Bad Request  | <a href="#">Notification400ErrorResponse</a>              |
|        |              | <pre>{\n  \"error\": \"invalid_notification_id\"\n}</pre> |
| 401    | Unauthorized |   |

## Schemas

### Notification400ErrorResponse

| Name  | Type   | Required | Description   | Schema |
|-------|--------|----------|---|--------|
| error | string | true     | An error code - must be <code>invalid_notification_request</code> , |        |

| Name              | Type   | Required | Description   | Schema |
|-------------------|--------|----------|---|--------|
|                   |        |          | <i>invalid_notification_id</i> , or<br><i>invalid_request</i> . |        |
| error_description | string | false    | A human-readable explanation of the error.                      |        |

This page was set to be reviewed before 19 August 2025. This might mean the content is out of date.



## Accessibility

## **OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Notification

The credential issuer notification endpoint is an optional endpoint defined in the [OID4VCI](#) ([https://openid.net/specs/openid-4-verifiable-credential-issuance-1\\_0.html#name-notification-endpoint](https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html#name-notification-endpoint)) specification. GOV.UK Wallet uses this endpoint to notify the credential issuer of events relating to issued credentials.

Notifications can be sent about events such as:

- the credential has been successfully stored in the GOV.UK Wallet
- the credential could not be stored in the GOV.UK Wallet because it is invalid
- the user has taken specific actions related to the offered credential, such as declining to save it

In order for GOV.UK Wallet to send notifications, the credential issuer must include a unique `notification_id` parameter in each [credential response \(/credential-issuer-functionality/credential/#credential-response\)](#).

## Technical details

### Endpoint location

The notification endpoint's location is implementation-specific within the OID4VCI specification.

The credential issuer must publish the location of their notification endpoint in their [metadata \(/credential-issuer-functionality/metadata/api.html#well-known-openid-credential-issuer\)](#) using the `notification_endpoint` parameter.

### Request format

The notification endpoint must accept HTTP POST requests. The request must include the:

- authorization header: A bearer access token (JWT) issued by GOV.UK One Login
- request body (JSON): The notification details

# Request validation

## Authorization header

The request must include an access token issued by the GOV.UK One Login token service as a bearer token in the Authorization header. This is the same as the access token used for authorising the issuance of a verifiable credential.

The credential issuer must validate the access token to ensure that it was issued by GOV.UK One Login and the request originates from the expected user.

To validate the access token, you should complete the following steps.

### 1. Verify the signature:

- retrieve the GOV.UK One Login token service's JSON Web Key Set (JWKS) from their published JWKS endpoint
- extract the `kid` (key ID) parameter from the access token header
- find the matching public key in the JWKS by comparing `kid` values
- confirm the `alg` (algorithm) parameter in the token header matches the algorithm of the identified public key
- use the matching public key to cryptographically verify the token signature using the specified algorithm

### 2. Validate the header parameters by ensuring that:

- the value of the `typ` (Type) parameter is "`at+jwt`"

### 3. Validate the payload claims by ensuring that:

- the value of the `iss` (issuer) claim matches the GOV.UK One Login URL:  
`"https://token.integration.account.gov.uk"` (integration) or  
`"https://token.account.gov.uk"` (production)
- the value of the `aud` (audience) claim is credential issuer URL
- the value of the `sub` (subject - this is the wallet subject identifier) claim matches the value stored in your cache for this specific credential issuance flow
- the value of the `exp` (expiration time) claim is in the future
- the value of the `credential_identifiers` claim matches the value stored in your cache for this specific credential issuance flow
- this API has not received another access token with the same `jti` (JWT ID) that is still within its validity period

## Request body

The request body must be in JSON format and contain the following parameters:

| Parameter         | Description   | Value(s)   |
|-------------------|---|--|
| notification_id   | A string (this could be a UUID) received in the credential response, uniquely identifying an individual credential issuance occurrence. |  |
| event             | A case-sensitive string indicating the credential's status.   | One of the following enums:<br><code>credential_accepted</code><br>(GOV.UK Wallet accepted the credential)<br><code>credential_failure</code><br>(GOV.UK Wallet rejected the credential)<br><code>credential_deleted</code><br>(user declined or deleted the credential) |
| event_description | An optional parameter that GOV.UK Wallet may include to provide additional information about the event.                                 |  |

The credential issuer must validate the contents of the request body and should ignore any unrecognised parameters.

Below is an example of a notification when a credential is successfully stored in the GOV.UK Wallet:

```
{  
  "notification_id": "776aefd4-26c6-4a5f-aa7c-b5e294cd87cd",  
  "event": "credential_accepted",  
  "event_description": "Credential has been successfully stored"  
}
```

More information about the notification request can be found in the [OID4VCI specification](#) ([https://openid.net/specs/openid-4-verifiable-credential-issuance-1\\_0.html#section-10.1](https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html#section-10.1)).

## Successful response format

When a notification is processed successfully, the credential issuer must return a 204 No Content HTTP status code.

## Error response format

If the notification could not be processed successfully, the credential issuer must return an appropriate HTTP error status code.

When the notification request does not include an access token or the access token is invalid, the notification endpoint must return a 401 Unauthorized HTTP status code and include the `WWW-Authenticate` response header field as defined in [RFC6750](https://datatracker.ietf.org/doc/html/rfc6750#section-3) (<https://datatracker.ietf.org/doc/html/rfc6750#section-3>), specifying the `Bearer` authentication scheme.

This is an example error response to a request with no access token:

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Bearer
Cache-Control: no-store
```

This is an example error response to an authentication attempt using an invalid access token:

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Bearer error="invalid_token"
Cache-Control: no-store
```

When there is an issue with the request body (e.g. the `notification_id` value is invalid), the notification endpoint must return a 400 Bad Request HTTP status code and a response body in JSON format containing the following parameter:

- `error` : A case-sensitive string indicating the error - `invalid_notification_id` (the request's `notification_id` was invalid), or `invalid_notification_request` (the request was invalid)

Below is an example of a notification error response when the notification request included an invalid `notification_id` :

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
Cache-Control: no-store
```

```
{  
  "error": "invalid_notification_id"  
}
```

More information about the error notification responses can be found in the [OID4VCI specification](#) ([https://openid.net/specs/openid-4-verifiable-credential-issuance-1\\_0.html#section-10.3](https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html#section-10.3)).

## Idempotency

The notification endpoint must be implemented idempotently. Identical requests with the same `notification_id` should always return the same response.

This page was last reviewed on 25 August 2025. It needs to be reviewed again on 25 February 2026 .



## Accessibility

### **OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Credential Issuer functionality

In this section:

- [Authenticate users with One Login](#)
- [Generate a credential offer](#)
- [Publish your metadata](#)
- [Publish your JSON Web Key Set \(JWKS\)](#)
- [Issue a credential](#)
- [Publish your DID document](#)
- [Handle notifications from GOV.UK Wallet](#)

This page was set to be reviewed before 5 September 2025. This might mean the content is out of date.



[Accessibility](#)

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Issuing credentials to GOV.UK Wallet

GOV.UK Wallet will support multiple credential formats to represent government documents. These documents can be:

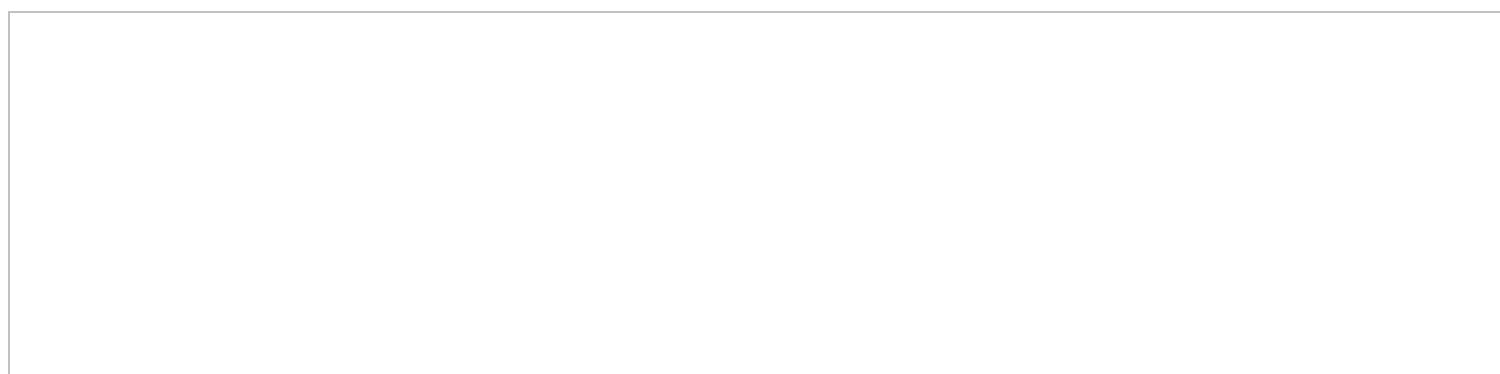
- [mdoc based credentials for the digital driving licence](https://www.iso.org/obp/ui/en/#iso:std:iso-iec:18013:-5:ed-1:v1:en)  
(<https://www.iso.org/obp/ui/en/#iso:std:iso-iec:18013:-5:ed-1:v1:en>)
- other Verifiable Credentials (VCs), including [W3C Verifiable Credential Data Model 2.0](https://www.w3.org/TR/vc-data-model-2.0/)  
(<https://www.w3.org/TR/vc-data-model-2.0/>) and later [other formats allowing selective disclosure of attributes](https://datatracker.ietf.org/doc/draft-ietf-oauth-sd-jwt-vc/) (<https://datatracker.ietf.org/doc/draft-ietf-oauth-sd-jwt-vc/>)

GOV.UK Wallet follows the [OpenID Connect for Verifiable Credential Issuance \(OIDC4VCI\)](https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html) ([https://openid.net/specs/openid-4-verifiable-credential-issuance-1\\_0.html](https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html)) open standard for its issuance flow.

Your team or department can start issuing credentials to GOV.UK Wallet by following this documentation.

## Understand GOV.UK Wallet's credential exchange flow

This diagram shows the exchange of a credential between a government service and GOV.UK Wallet. Below the diagram is an explanation of each step in the process.



## User authenticates with GOV.UK One Login to use your service (/credential-issuer-functionality/authenticating-users/#authenticating-users-with-one-login)

1. Your user accesses your service.
2. Your service authenticates the user with GOV.UK One Login.
3. Your service fetches the user's `walletSubjectId` from the GOV.UK One Login [/userinfo API \(https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/authenticate-your-user/#retrieve-user-information\)](#).

There is detailed guidance on [how GOV.UK One Login works \(https://docs.sign-in.service.gov.uk/how-gov-uk-one-login-works/\)](#) in the GOV.UK One Login technical documentation.

## Your service issues a credential offer (/credential-issuer-functionality/credential-offer/#credential-offer)

4. Your service generates a credential offer. Included in this offer is a [pre-authorised code \(/credential-issuer-functionality/credential-offer/#the-pre-authorised-code\)](#) signed by your service.
5. Your service renders the credential offer to the user as a QR code or deep-link.
6. The user opens the app.
7. The app prompts the user to authenticate with GOV.UK One Login.
8. The user who authenticated with your service in a web browser is authenticated with GOV.UK One Login in the app.
9. The user scans the QR code or opens the deep link. This action passes the credential offer to GOV.UK Wallet.

## Your service publishes metadata about the credentials it publishes

10. GOV.UK Wallet sends a GET request to your `/.well-known/openid-credential-issuer` endpoint to fetch your metadata.
11. Your service returns its metadata.
12. GOV.UK Wallet calls GOV.UK One Login to exchange the pre-authorised code in the credential offer for an access token.
13. GOV.UK One Login sends a GET request to your `/.well-known/jwks.json` endpoint to fetch your public keys, which verify the signature on the pre-authorised code issued by your service.
14. Your service returns its public keys as a [JSON Web Key Set \(JWKS\)](#).
15. GOV.UK One Login verifies the pre-authorised code content and its signature.

16. GOV.UK One Login issues an access token that you can trust when GOV.UK Wallet calls your service to redeem it.
17. GOV.UK Wallet generates a [proof of possession](#) for the key material.
18. GOV.UK Wallet sends a POST request to your [/credential](#) endpoint to request the credential. This request includes the access token issued by GOV.UK One Login (as a bearer token in the authorization header) and the proof of possession generated by GOV.UK Wallet.

## [Your service issues a credential \(/credential-issuer-functionality/credential/#credential\)](#)

19. Your service sends a GET request to the GOV.UK One Login [/.well-known/jwks.json](#) to fetch its public keys, which verify the signature on the access token issued by GOV.UK One Login.
20. GOV.UK One Login returns its public keys as a JSON Web Key Set (JWKS).
21. Your service verifies the content and signature of the access token and the proof of possession.
22. Your service compares the [walletSubjectId](#) in the access token's [sub](#) claim with the [walletSubjectId](#) retrieved in step 3. If they are the same, this provides assurance that you are issuing the credential to a digital wallet that is logged in as the rightful holder.
23. Your service builds and signs the credential, and [binds it to the did:key provided in the proof of possession](#) to make sure the credential can only be used by the device it is issued to.
24. Your service returns the device-bound credential to GOV.UK Wallet.
25. GOV.UK Wallet sends a GET request to your [/.well-known/did.json](#) endpoint to fetch your [DID document](#). The DID document contains your public key which is required to verify the signature on the credential issued by your service.
26. Your service returns its DID document.
27. GOV.UK Wallet verifies the content and signature of the credential.
28. GOV.UK Wallet stores the credential.

## [GOV.UK Wallet notifies your service \(/credential-issuer-functionality/notification/\)](#)

The following steps are optional. If you do not offer a [/notification](#) endpoint then GOV.UK Wallet will not send a notification.

29. GOV.UK Wallet sends a POST request to your [/notification](#) endpoint to notify your service. This notification will confirm whether GOV.UK Wallet successfully stored the credential, or failed to store it.
30. Your service records the notification.
31. Your service returns an empty response to GOV.UK Wallet.

This page was last reviewed on 14 May 2025. It needs to be reviewed again on 14 November 2025 .



## Accessibility

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Key Management

GOV.UK Wallet needs to verify the validity of the credentials your service issues.

When issuing credentials in [W3C Verifiable Credential Data Model 2.0](#) (<https://www.w3.org/TR/vc-data-model-2.0/>) format and signing with your private keys, your credentials need to be verified by the public keys you made available in the did:web document.

Your public keys need to stay available through the lifecycle of your credentials. A public key used to sign a group of verifiable credentials (VCs) can not be made inactive until after the VCs have expired. Public keys should be kept in an inactive state, available to be verified.

For their credential issuer service, credential issuers should include specific key management features.

The service needs a key refresh process that creates a new asymmetric public or private key pair for signing new VCs, but that retains trust in the previous versions of the public key for verifying.

This is done by making sure the public part of the historical key is retained, while the private key is destroyed. For example, a VC issued by an internal and external issuer.

The service also needs key revocation. This needs to include a notice made from the credential issuer explaining that a specific key should be removed from operational use before the key expires. This will generally happen when the key is lost or compromised. If a key is compromised, it can be used by an attacker to decrypt or forge messages, impersonate an identity, or access sensitive information.

The table below describes the possible states of a key pair used for signing credentials:

| Key State | Description   |
|-----------|---|
| Created   | A key pair is generated with an activation date in the future. It is not yet used for signing.  |
| Active    | A key becomes active on the activation date, and enabled for signing and verifying the VC. There must not be multiple keys active at the same time. |

|          |  |
|----------|--|
| Inactive | A key becomes inactive past its expiration date or time. The public key will still be valid for verifying the VC.  |
| Revoked  | A key is destroyed and removed from the issuer's server, and is not valid for signing or verifying the signatures. |

This page was set to be reviewed before 5 September 2025. This might mean the content is out of date.



## Accessibility

### **OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Before you issue a status record

To issue credentials with a status, you must register with the Status List Service. To do this, speak to your GOV.UK Wallet engagement manager or [contact us \(/contact-us.html\)](#).

## Register a status list client

Only valid status list clients can issue and revoke statuses in the Status List Service.

When registering as a status list client, you must provide:

- a public JSON web key set (JWKS) endpoint - used to verify the signatures of the status list client's signed JSON web tokens (JWTs) for the issue and revoke endpoint
- a status list type (Bitstring or Token) - the type of status issued for this specific status list client

You can register multiple status list clients if you need to.

When you complete your registration, you get a unique `clientId`. You must include your `clientId` identifier as [the `iss` claim \(<https://datatracker.ietf.org/doc/html/rfc7519#section-4.1.1>\)](https://datatracker.ietf.org/doc/html/rfc7519#section-4.1.1) in the JWTs you send to the `/issue` and `/revoke` endpoints.

The requests to and responses from the Status List Service are the same, regardless of the status list type. You must make sure the Status List Service's responses are included in your credential correctly.

## Access the Status List Service APIs

When you register as a status list client, we will work with you to grant access to our APIs and provide the API URLs.

## Test your integration

The Status List Service operates an integration and a production environment. You must register a status list client for each environment.

We recommend that you use the integration environment to test your integration with the Status List Service before you move to production. The integration environment should not be used for publicly issued credentials.

This page was last reviewed on 22 October 2025. It needs to be reviewed again on 22 April 2026 .



## Accessibility

## **OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# API

## /issue

### post

An endpoint to request a new status index for a GOV.UK wallet credential. This API will use the issuer's `/.well-known/jwks.json` endpoint to locate the signing keys to verify the signature.

#### Responses

| Status | Description  | Schema                                |
|--------|--|---------------------------------------|
| 200    | A response that contains a status index and a uri to the status list where this newly issued <b>VALID</b> (binary 00) status index value can be found. | <a href="#">IssueResponse</a>         |
|        | <pre>{<br/>  "index": 3,<br/>  "uri": "https://crs.account.gov.uk/b/A671FED3E9AD"<br/>}</pre>  |                                       |
| 400    | Bad request (invalid JWT, missing fields, wrong content-type, etc.)  | <a href="#">Issue400ErrorResponse</a> |
|        | <pre>{<br/>  "error": "BAD_REQUEST",<br/>  "error_description": "No Type in Header"<br/>}</pre>  |                                       |
| 401    | Unauthorized (client not found)  | <a href="#">Issue401ErrorResponse</a> |
|        | <pre>{<br/>  "error": "UNAUTHORISED",<br/>  "error_description": "No matching client found with ID: invalidClientId"<br/>}</pre>                       |                                       |
| 403    | Forbidden (JWT signature verification failure)   | <a href="#">Issue403ErrorResponse</a> |
|        | <pre>{<br/>  "error": "FORBIDDEN",<br/>  "error_description": "Failure verifying the signature of the jwt"<br/>}</pre>                                 |                                       |
| 500    | Internal server error  | <a href="#">Issue500ErrorResponse</a> |

| Status | Description | Schema  |
|--------|-------------|---|
|        |             | <pre>{   "error": "INTERNAL_SERVER_ERROR",   "error_description": "Error receiving messages: ..." }</pre> |

## Schemas

### IssueResponse

| Name | Type   | Required | Description   | Schema |
|------|--------|----------|---|--------|
| idx  | number | true     | The assigned status list index. This index position is unique within the status list identified by the URI                              |        |
| uri  | string | true     | The URI of the status list that holds the issued credential. It is used in combination with the index to retrieve the status list entry |        |

### Issue400ErrorResponse

| Name              | Type   | Required | Description                                  | Schema |
|-------------------|--------|----------|--|--------|
| error             | string | true     | An error code - must be <b>BAD_REQUEST</b> . |        |
| error_description | string | false    | A human-readable explanation of the error.   |        |

### Issue401ErrorResponse

| Name              | Type   | Required | Description                                   | Schema |
|-------------------|--------|----------|---|--------|
| error             | string | true     | An error code - must be <b>UNAUTHORISED</b> . |        |
| error_description | string | false    | A human-readable explanation of the error.    |        |

### Issue403ErrorResponse

| Name              | Type   | Required | Description                                | Schema |
|-------------------|--------|----------|--|--------|
| error             | string | true     | An error code - must be <b>FORBIDDEN</b> . |        |
| error_description | string | false    | A human-readable explanation of the error. |        |

### Issue500ErrorResponse

| Name              | Type   | Required | Description  | Schema |
|-------------------|--------|----------|--|--------|
| error             | string | true     | An error code - must be <b>INTERNAL_SERVER_ERROR</b> . |        |
| error_description | string | false    | A human-readable explanation of the error.             |        |

This page was last reviewed on 22 October 2025. It needs to be reviewed again on 22 April 2026.



## Accessibility

### **OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Issue a status list entry

You can use the `/issue` endpoint to request a status list slot for a credential you want to issue. This endpoint lets you retrieve a new `uri / idx` pair from a status list, which represents a unique credential being issued to a user's wallet.

To use the `/issue` endpoint you must [register as a credential issuer with the Status List service \(/status-list/before-issuing-status-record\)](#), and you must send your request as a signed JSON Web Token (JWT).

When you request a new `uri / idx` pair for a credential on a status list, the Status List service validates your request. If this validation is successful, the Status List service will issue a new `uri / idx` pair on one of its status lists.

The Status List service will also return a uri and an index to where you can retrieve this credential's status. You must include this response in your issued credential.

For guidance on how to include the Status List Service's response in your credentials, see:

- the [Token status list specification \(<https://datatracker.ietf.org/doc/draft-ietf-oauth-status-list/>\)](#) for mdoc credentials
- the [BitString status list specification \(<https://www.w3.org/TR/vc-bitstring-status-list/>\)](#) for JWT-VC credentials

## Technical details

The requests to and responses from the Status List Service are the same, regardless of the credential or status list type.

### Endpoint URI

The URI path for the issue endpoint is `/issue`.

When you register as a credential issuer with the Status List service, you get access to the internal API. You must sign the request with your private key and share public keys on your `/.well-known/jwks.json` endpoint. This is used to verify the JWT.

### Request format

The issue endpoint only accepts HTTP POST requests.

The request must include:

- **header** : you must provide the `Content-Type` header - the only valid value is `application/jwt`
- **request body (JWT)** : contains a signed JWT based on [RFC 7515](#) (<https://datatracker.ietf.org/doc/html/rfc7515>), which must follow the requirements below

## Status list JWT definition /issue

### Header

The JSON Object Signing and Encryption (JOSE) header (based on [RFC-7515](#) (<https://datatracker.ietf.org/doc/html/rfc7515#section-4>)) must contain the following header parameters:

```
{  
  "typ": "JWT",  
  "alg": "ES256",  
  "kid": "1fb2c0f07f643b45cafeb53fb9d9eb34"  
}
```

| Parameter        | Required or optional | Description   |
|------------------|----------------------|---|
| <code>typ</code> | Required             | <code>typ</code> stands for ‘type’. You must set this value to be <code>JWT</code> . This is the media type of the complete JWT.              |
| <code>alg</code> | Required             | <code>alg</code> stands for ‘algorithm’. You must set this value to be <code>ES256</code> . This is the algorithm used to sign the JWT.       |
| <code>kid</code> | Required             | <code>kid</code> stands for ‘key ID’. This key ID must be present in your hosted JWKS. This is used to validate the JSON web signature (JWS). |

### Payload

The JWT payload must contain the following claims:

```
{  
  "iss": "exampleclientIDabcd123",
```

```
"iat": 1686920170,  
"jti": "62b45850-4c5c-4696-983a-af66450301d4",  
"statusExpiry": 1734709493  
}
```

| Claim         | Required or optional | Description  |
|---------------|----------------------|--|
| iss           | Required             | <p><code>iss</code> stands for ‘issuer’. This is the <a href="#">clientId of the credential issuer (/status-list/before-issuing-status-record)</a> service generated when registering as a client.</p> <p>Make sure you are using the correct <code>clientId</code> for your environment - production or integration.</p>  |
| iat           | Required             | <p><code>iat</code> stands for ‘issued at’. This is the UNIX timestamp when the request JWT was issued.</p>  |
| jti           | Required             | <p><code>jti</code> stands for ‘JWT ID’. This provides a unique identifier for the JWT. The Status List Service will validate the format provided to make sure it is a lowercase UUID.</p>   |
| status Expiry | Required             | <p>The point after which the status expires. After this date the credential will be removed from the status list.</p> <p><code>statusExpiry</code> must be equal to or later than the issued credential’s technical expiry time, known as the <code>validUntil</code> property. It may be useful to issue a status before the <code>validUntil</code> value is known. In this case, we recommend that you keep any resulting difference between <code>validUntil</code> and <code>statusExpiry</code> to a minimum.</p> <p><code>statusExpiry</code> must be a number in seconds, formatted as a UNIX timestamp. The Status List Service does not support credentials lasting over 10 years.</p> |

## Example Request

Below is an example of the post request signed and encoded as a JWT.

```
POST /issue HTTP/1.1  
Host: <API.CRS.ACCOUNT.GOV.UK>  
Content-Type: application/jwt
```

Accept: application/json

eyJhbGciOiJFUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6IjgwODY4Nzk0LTM2MjYtNDNmOC05YTRk

## Example Response

HTTP/1.1 200 OK

Content-Type: application/json

```
{  
  "uri": "https://crs.account.gov.uk/b/A671FED3E9AD"  
  "idx": 3,  
}
```

### Parameter Description

|            |   |
|------------|---|
| <b>uri</b> | <b>uri</b> stands for ‘uniform resource identifier’. This is the uri to the status list endpoint in which the new credential has been stored.<br><br>This will be formatted as <code>crs.account.gov.uk</code> for the production environment, and <code>crs.integration.account.gov.uk</code> for the integration environment. |
| <b>idx</b> | <b>idx</b> stands for ‘index’. This is the index at which the credential will be stored in the status list found on the uri.  |

This page was last reviewed on 22 October 2025. It needs to be reviewed again on 22 April 2026 .





[Table of contents](#)

# API

## /revoke

### post

An endpoint used by GOV.UK Wallet credential issuers to revoke a status list index associated with a GOV.UK Wallet credential that they issued. This API will use the issuer's `/.well-known/jwks.json` endpoint to locate the signing keys to verify the signature.

### Responses

| Status | Description  | Schema                                 |
|--------|--|--|
| 202    | Revocation processed successfully  | <a href="#">RevokeResponse</a>         |
|        | <pre>{<br/>  "message": "Request processed for revocation",<br/>  "revokedAt": 1734709493<br/>}</pre>                            |  |
| 400    | Bad request (invalid JWT, missing fields, wrong content-type, etc.)  | <a href="#">Revoke400ErrorResponse</a> |
|        | <pre>{<br/>  "error": "BAD_REQUEST",<br/>  "error_description": "No Type in Header"<br/>}</pre>                                  |  |
| 401    | Unauthorized (client not found or client mismatch)   | <a href="#">Revoke401ErrorResponse</a> |
|        | <pre>{<br/>  "error": "UNAUTHORISED",<br/>  "error_description": "No matching client found with ID: invalidClientId"<br/>}</pre> |  |
| 403    | Forbidden (JWT signature verification failure or JWKS fetch failure)   | <a href="#">Revoke403ErrorResponse</a> |
|        | <pre>{<br/>  "error": "FORBIDDEN",<br/>  "error_description": "Failure verifying the signature of the jwt"<br/>}</pre>           |  |
| 404    | Entry not found or list type mismatch  | <a href="#">Revoke404ErrorResponse</a> |

**Status Description****Schema**

```
{  
    "error": "NOT_FOUND",  
    "error_description": "Entry not found in status list table"  
}
```

500 Internal server error

[Revoke500ErrorResponse](#)

```
{  
    "error": "INTERNAL_SERVER_ERROR",  
    "error_description": "Error processing revocation request"  
}
```

## Schemas

### RevokeResponse

| Name      | Type   | Required | Description                                    | Schema |
|-----------|--------|----------|--|--------|
| message   | string | true     | Status message indicating the result           |        |
| revokedAt | number | true     | Unix timestamp when the credential was revoked |        |

### Revoke400ErrorResponse

| Name              | Type   | Required | Description                                  | Schema |
|-------------------|--------|----------|--|--------|
| error             | string | true     | An error code - must be <b>BAD_REQUEST</b> . |        |
| error_description | string | false    | A human-readable explanation of the error.   |        |

### Revoke401ErrorResponse

| Name              | Type   | Required | Description                                   | Schema |
|-------------------|--------|----------|---|--------|
| error             | string | true     | An error code - must be <b>UNAUTHORISED</b> . |        |
| error_description | string | false    | A human-readable explanation of the error.    |        |

### Revoke403ErrorResponse

| Name  | Type   | Required | Description                                | Schema |
|-------|--------|----------|--|--------|
| error | string | true     | An error code - must be <b>FORBIDDEN</b> . |        |

| Name              | Type   | Required | Description                                | Schema |
|-------------------|--------|----------|--|--------|
| error_description | string | false    | A human-readable explanation of the error. |        |

## Revoke404ErrorResponse

| Name              | Type   | Required | Description                                | Schema |
|-------------------|--------|----------|--|--------|
| error             | string | true     | An error code - must be <b>NOT_FOUND</b> . |        |
| error_description | string | false    | A human-readable explanation of the error. |        |

## Revoke500ErrorResponse

| Name              | Type   | Required | Description  | Schema |
|-------------------|--------|----------|--|--------|
| error             | string | true     | An error code - must be <b>INTERNAL_SERVER_ERROR</b> . |        |
| error_description | string | false    | A human-readable explanation of the error.             |        |

This page was last reviewed on 22 October 2025. It needs to be reviewed again on 22 April 2026 .



## Accessibility

**OGL** All content is available under the [Open Government Licence v3.0](#), except where otherwise stated © Crown copyright

[Table of contents](#)

# Revoke a credential

As a credential issuer, you can call the `/revoke` endpoint to revoke a credential you have previously issued. You cannot use it to revoke a credential issued by anyone else.

The Status List Service validates all calls to the `/revoke` endpoint to make sure that the caller has the correct rights to revoke the credential.

When you call the `/revoke` endpoint on an existing status, the status list records that status as revoked. This state change will be reflected in the published status list within a short timeframe. This process can not be reversed. There is [more guidance on this in the statuslist endpoint page](#).

## Technical details

The requests to and responses from the Status List Service are the same, regardless of the credential or status list type.

### Endpoint URI

The URI path for the revoke credential endpoint is `/revoke` .

When you register as a credential issuer with the Status List Service, you get access to the internal API. You must sign the request with your private key and share public keys on your `/.well-known/jwks.json` endpoint. This is used to verify the JSON web token (JWT).

### Request format

The revoke endpoint only accepts HTTP POST requests.

The request must include:

- `header` : you must provide the `Content-Type` header - the only valid value is `application/jwt`
- `request body` : contains a signed JWT based on [RFC 7515](#) (<https://datatracker.ietf.org/doc/html/rfc7515>), which must follow the requirements below

# Status list JWT definition / revoke

## Header

The JSON Object Signing and Encryption (JOSE) header (based on [RFC-7515](#) (<https://datatracker.ietf.org/doc/html/rfc7515#section-4>)) must contain the following header parameters:

```
{  
  "typ": "JWT",  
  "alg": "ES256",  
  "kid": "499b46712489a805510bdf3e61e1f93d"  
}
```

| Parameter | Required or optional | Description  |
|-----------|----------------------|--|
| typ       | Required             | typ stands for ‘type’. You must set this value to be <a href="#">JWT</a> . This is the media type of the complete JWT.           |
| alg       | Required             | alg stands for ‘algorithm’. You must set this value to be <a href="#">ES256</a> . This is the algorithm used to sign the JWT.    |
| kid       | Required             | kid stands for ‘key ID’. This key ID must be present in your hosted JWKS. This is used to validate the JSON web signature (JWS). |

## Payload

The JWT payload must contain the following claims:

```
{  
  "iss": "asKWnsjeEJEWjjwSHsIksIksIhBe",  
  "iat": 1686920170,  
  "jti": "62b45850-4c5c-4696-983a-af66450301d4",  
  "uri": "https://crs.account.gov.uk/t/3B0F3BD087A7",  
  "idx": 3  
}
```

| Claim | Required or optional | Description   |
|-------|----------------------|---|
| iss   | Required             | <p><code>iss</code> stands for ‘issuer’. This is the <a href="#">clientId of the credential issuer (/status-list/before-issuing-status-record)</a> service generated when registering as a client.</p> <p>Make sure you are using the correct <code>clientId</code> for your environment - production or integration.</p> |
| iat   | Required             | <code>iat</code> stands for ‘issued at’. This is the UNIX timestamp when the request JWT was issued.  |
| jti   | Required             | <code>jti</code> stands for ‘JWT ID’. This provides a unique identifier for the JWT. The Status List Service will validate the format provided to make sure it is a lowercase UUID.   |
| uri   | Required             | <code>uri</code> stands for ‘uniform resource identifier’. This is the uri of the status list that holds the status to revoke.  |
| idx   | Required             | <code>idx</code> stands for ‘index’. This is the index of the status to be revoked.   |

Your `uri` and `idx` must exactly match the response from the [/issue endpoint \(/status-list/issue-status-list-entry\)](#).

## Example Request

```
POST /revoke HTTP/1.1
Host: <API.CRS.ACCOUNT.GOV.UK>
Content-Type: application/jwt

eyJhbGciOiJFUzI1NiIsInR5cCI6IkpxVCIsImtpZCI6IjgwODY4Nzk0LTM2MjYtNDNmOC05YTRk
```

## Example Response

```
HTTP/1.1 202 ACCEPTED
```

This page was last reviewed on 22 October 2025. It needs to be reviewed again on 22 April 2026 .



## Accessibility

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Bitstring status list API

## /b/{statusListIdentifier}

### get

A public endpoint that returns a W3C Bitstring Status List credential JWT. The list that is returned depends on the list type for the GOV.UK Wallet credential issuer agreed at the time of registration. The response has a signed JWT in one of the two supported status list formats.

### Parameters

| Parameter            | In   | Type   | Required | Description                                       |
|----------------------|------|--------|----------|---|
| statusListIdentifier | path | string | true     | Unique name representing the specific status list |

### Responses

| Status | Description           | Schema   |
|--------|-----------------------|--|
| 200    | OK                    |  |
| 404    | Not Found             | <a href="#">StatusList404ErrorResponse</a>   |
|        |                       | <pre>{<br/>  "error": "NOT_FOUND",<br/>  "error_description": "Status List not found for endpoint uri"<br/>}</pre> |
| 500    | Internal server error | <a href="#">StatusList500ErrorResponse</a>   |
|        |                       | <pre>{<br/>  "error": "INTERNAL_SERVER_ERROR",<br/>  "error_description": "..."<br/>}</pre>                        |

### Schemas

## StatusList404ErrorResponse

| Name              | Type   | Required | Description                                      | Schema |
|-------------------|--------|----------|--|--------|
| error             | string | true     | An error code - must be <code>NOT_FOUND</code> . |        |
| error_description | string | false    | A human-readable explanation of the error.       |        |

## StatusList500ErrorResponse

| Name              | Type   | Required | Description  | Schema |
|-------------------|--------|----------|--|--------|
| error             | string | true     | An error code - must be <code>INTERNAL_SERVER_ERROR</code> . |        |
| error_description | string | false    | A human-readable explanation of the error.                   |        |

This page was last reviewed on 22 October 2025. It needs to be reviewed again on 22 April 2026 .



### Accessibility

### **OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# JSON Web Key Set (JWKS) API

The Status List Service's JSON Web Key Set (JWKS) endpoint. This provides a list of public keys which can be used to verify the signature on JWTs provided by the Status List Service.

## ./well-known/jwks.json

### get

A public endpoint that stores the JSON Web Key Set (JWKS) of public keys issued by a service. These keys can be used by client applications to verify the signature of a JSON Web Token (JWT).

### Responses

| Status | Description               | Schema                       |
|--------|---------------------------|------------------------------|
| 200    | Credential issuer's JWKS. | <a href="#">JwksResponse</a> |

```
{  
  "keys": [  
    {  
      "kty": "EC",  
      "use": "sig",  
      "crv": "P-256",  
      "kid": "5dcbee863b5d7cc30c9ba1f7393dacc6c16610782e4b6a191f94a7e8b1e1510f",  
      "x": "6jCKX_QRrmTeEJi-uiwcYqu8BgMg170g2pdAst24MPE=",  
      "y": "icPzjbSk6apD_SNvQt8NW0PlPeGG4KYU55GfnARryoY=",  
      "alg": "ES256"  
    }  
  ]  
}
```

## Schemas

### JwksResponse

| Name | Type  | Required | Description  | Schema              |
|------|-------|----------|--|---------------------|
| keys | array | true     | A set of public keys, each in JSON Web Key (JWK) format. | <a href="#">Key</a> |

### Key

| Name | Type   | Required | Description   | Schema |
|------|--------|----------|---|--------|
| kty  | string | true     | Key Type. The family of cryptographic algorithms used with the key. |        |
| kid  | string | true     | Key ID. Unique identifier to match a specific key.                  |        |
| crv  | string | true     | Curve. Cryptographic curve used with the key.                       |        |
| x    | string | true     | The “x” coordinate for the elliptic curve point.                    |        |
| y    | string | true     | The “y” coordinate for the elliptic curve point.                    |        |
| alg  | string | true     | Algorithm. The cryptographic algorithm used with the key.           |        |
| use  | string | true     | The intended use of the key.  |        |

This page was last reviewed on 22 October 2025. It needs to be reviewed again on 22 April 2026 .



## Accessibility

## **OGL**

All content is available under [the Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Token status list API

## /t/{statusListIdentifier}

### get

A public endpoint that returns an IETF Token Status List JWT. The list that is returned depends on the list type for the GOV.UK Wallet credential issuer agreed at the time of registration. The response has a signed JWT in one of the two supported status list formats.

### Parameters

| Parameter            | In   | Type   | Required | Description                                       |
|----------------------|------|--------|----------|---|
| statusListIdentifier | path | string | true     | Unique name representing the specific status list |

### Responses

| Status | Description           | Schema   |
|--------|-----------------------|--|
| 200    | OK                    |  |
| 404    | Not Found             | <a href="#">StatusList404ErrorResponse</a>   |
|        |                       | <pre>{<br/>  "error": "NOT_FOUND",<br/>  "error_description": "Status List not found for endpoint uri"<br/>}</pre> |
| 500    | Internal server error | <a href="#">StatusList500ErrorResponse</a>   |
|        |                       | <pre>{<br/>  "error": "INTERNAL_SERVER_ERROR",<br/>  "error_description": "..."<br/>}</pre>                        |

### Schemas

## StatusList404ErrorResponse

| Name              | Type   | Required | Description                                      | Schema |
|-------------------|--------|----------|--|--------|
| error             | string | true     | An error code - must be <code>NOT_FOUND</code> . |        |
| error_description | string | false    | A human-readable explanation of the error.       |        |

## StatusList500ErrorResponse

| Name              | Type   | Required | Description  | Schema |
|-------------------|--------|----------|--|--------|
| error             | string | true     | An error code - must be <code>INTERNAL_SERVER_ERROR</code> . |        |
| error_description | string | false    | A human-readable explanation of the error.                   |        |

This page was last reviewed on 22 October 2025. It needs to be reviewed again on 22 April 2026 .



### Accessibility

### **OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Check a credential's status

The Status List Service hosts all status lists publicly. Each list is a signed JSON web token (JWT). You can verify the JWT's signature by accessing [the Status List Service's JSON web key set \(JWKS\)](#) hosted at <https://crs.account.gov.uk/.well-known/jwks.json> (production) or <https://crs.integration.account.gov.uk/.well-known/jwks.json> (integration).

There are two formats that the Status List Service supports: [Bitstring status lists](#) and [Token status lists](#).

You can use the status list `uri` in the credential to check a credential's status.

Each status at a specific index in the status list uses 2 bits. Each status index may contain one of the following bit combinations:

- `00` (VALID). Represents a valid credential
- `01` (INVALID). Represents a credential that has been permanently revoked (marked as invalid)
- `10` (NOT USED). Currently not used
- `11` (NOT USED). Currently not used

## Bitstring status list

Status lists where the URI path begins with `/b/` are Bitstring status lists that follow the [W3C Bitstring Status List specification](#) (<https://www.w3.org/TR/vc-bitstring-status-list/>).

For consistency between the two different lists that the status list service publishes, the Status List Service uses the more [complex implementation of Bitstring status lists](#) (<https://www.w3.org/TR/vc-bitstring-status-list/#example-example-statuslistcredential-using-more-complex-entries>).

## Technical details

### Endpoint URI

The URI path for the Bitstring status list endpoint is `/b/{statusListIdentifier}`. It is presented as a GET request, where:

- **b** represents the type of status list: `BitstringStatusList`
- `statusListIdentifier` represents an ID for a specific status list

## Bitstring status list request example

Below is an example of the `/b/{statusListIdentifier}` request:

```
GET /b/A671FED3E9AD HTTP/1.1
Host: crs.account.gov.uk
Accept: application/json
```

## Request Response

### Header

The JWT response header will contain the following:

```
{
  "alg": "ES256",
  "kid": "12",
  "typ": "vc+jwt"
}
```

### Parameter Description

|                  |  |
|------------------|--|
| <code>alg</code> | <code>alg</code> stands for ‘algorithm’. This value will be returned as <code>ES256</code> . This is the algorithm used to encode the JWT.                     |
| <code>kid</code> | <code>kid</code> stands for ‘key ID’. This key ID represents a key in the Status List Service’s JWKS which can be used to verify the JSON web signature (JWS). |
| <code>typ</code> | <code>typ</code> stands for ‘type’. This is the type of the status list. It is <code>vc+jwt</code> for Bitstring status lists.                                 |

### Payload

The JWT response payload for a Bitstring status list will contain the following:

```
{
  "@context": [
    "https://www.w3.org/ns/credentials/v2",
    "https://www.w3.org/ns/credentials/examples/v2"
  ],
  "id": "https://crs.account.gov.uk/b/A671FED3E9AD",
  "type": [
    "VerifiableCredential",
    "BitstringStatusListCredential"
  ],
  "issuer": "https://crs.account.gov.uk/",
  "validFrom": "2025-10-01T14:00:00Z",
  "validUntil": "2025-10-08T14:00:00Z",
  "credentialSubject": {
    "id": "https://crs.account.gov.uk/b/A671FED3E9AD#list",
    "type": "BitstringStatusList",
    "statusSize": 2,
    "statusPurpose": "message",
    "statusMessage": [
      {
        "status": "0x0",
        "message": "VALID"
      },
      {
        "status": "0x1",
        "message": "INVALID"
      }
    ],
    "encodedList": "uH4sIAAAAAAAA3MUBABJTAvCAgAAAA",
    "ttl": "3600"
  }
}
```

| Parameter   | Description                                    |
|-------------|--|
| <b>id</b>   | A unique URL that represents this status list. |
| <b>type</b> | The type of credential.                        |

|   |   |
|---|---|
| <code>issuer</code>                                   | The URL of this status list credential's issuer.  |
| <code>validFrom</code>                                | The earliest point in time at which the status list is valid.   |
| <code>validUntil</code>                               | The latest point in time at which the status list is valid.   |
| <code>credentialSubject</code>                        | The status list subject about which the claims below are made.  |
| <code>credentialSubject.id</code>                     | A unique URI that represents this status list.  |
| <code>credentialSubject.type</code>                   | The type of credential. This will be <code>BitstringStatusList</code> .   |
| <code>credentialSubject.statusSize</code>             | The size of the status list in bits.  |
| <code>credentialSubject.statusPurpose</code>          | The purpose of the status list, as described in <code>statusMessage</code> .  |
| <code>credentialSubject.statusMessages</code>         | This is an array of objects, which each contain a status and a message.   |
| <code>credentialSubject.statusMessages.status</code>  | This represents the status value in the status list. It is a hexadecimal string, and will be <code>"0x0"</code> or <code>"1x1"</code> .   |
| <code>credentialSubject.statusMessages.message</code> | The status message representing the status value. This will be <code>"VALID"</code> or <code>"INVALID"</code> .   |
| <code>credentialSubject.encodedList</code>            | This is a multibase-encoded base64url (with no padding) representation of the GZIP-compressed bitstring values for the associated range of verifiable credential status values. |

## Token status list

Status lists where the URI path begins with `/t/` are Token status lists that follow the [IETF Token Status List specification](#) (<https://www.ietf.org/archive/id/draft-ietf-oauth-status-list-12.html>).

## Technical details

### Endpoint URI

The URI path for the Token status list endpoint is `/t/{statusListIdentifier}` . It is presented as a GET request, where:

- `t` represents the type of status list: `TokenStatusList`
- `statusListIdentifier` represents an ID for a specific status list

## Token Status List Request Example

Below is an example of the `/t/{statusListIdentifier}` request:

```
GET /t/A671FED3E9AD HTTP/1.1
Host: crs.account.gov.uk
Accept: application/statuslist+jwt
```

## Request Response

### Header

The JWT response header will contain the following:

```
{
  "alg": "ES256",
  "kid": "12",
  "typ": "statuslist+jwt"
}
```

## Parameter Description

---

**alg** `alg` stands for ‘algorithm’. This value will be returned as `ES256` . This is the algorithm used to encode the JWT.

---

**kid** `kid` stands for ‘key ID’. This key ID represents a key in the Status List Service’s JWKS which can be used to verify the JSON web signature (JWS).

---

**typ** `typ` stands for ‘type’. This is the type of the status list. This will be `statuslist+jwt` for a Token status list.

---

## Payload

The JWT response payload for a Token status list will contain the following:

```
{  
  "exp": 2291720170,  
  "iat": 1686920170,  
  "iss": "https://crs.account.gov.uk",  
  "status_list": {  
    "bits": 2,  
    "lst": "eNpzdAEAMgAhg"  
  },  
  "sub": "https://crs.account.gov.uk/b/A671FED3E9AD",  
  "ttl": 43200  
}
```

| Parameter                                      | Description  |
|--|--|
| <code>exp</code>                               | <code>exp</code> stands for ‘expiry’. This is the expiry of the subject credential.  |
| <code>iat</code>                               | <code>iat</code> stands for ‘issued at’. This is the timestamp the subject credential was originally issued at.  |
| <code>iss</code>                               | <code>iss</code> stands for ‘issuer’. This is the URL of the credential issuer service operated by the organisation sharing the credential.                        |
| <code>status_list.</code><br><code>bits</code> | The number of bits that represent a status.  |
| <code>status_list.</code><br><code>lst</code>  | <code>lst</code> stands for ‘list’. This is an encoded version of this status list.  |
| <code>sub</code>                               | <code>sub</code> stands for ‘subject’. This is the URI of the status list that was in the original HTTP request.   |
| <code>ttl</code>                               | <code>ttl</code> stands for ‘time-to-live’. This is the lifetime of the cached version of this status list. Status lists are updated at regular and set intervals. |

## JSON Web Key Set (JWKS)

The JWKS endpoint exposes the Status List Service’s public cryptographic keys in JSON Web Key Set (JWKS) format. You can use a public key to verify the signature of a status

list. This verification lets you make sure that the status list was published by the Status List Service and it has not been tampered with.

## Technical details

### Endpoint location

The JWKS is publicly accessible at the standardised location `/.well-known/jwks.json` on the Status List Service domain.

### Response format

The endpoint must return a 200 OK HTTP status code and a valid JSON response that follows the JWKS specification defined in [RFC 7517](#) (<https://datatracker.ietf.org/doc/html/rfc7517>). Each key within the JWKS is represented as a JSON Web Key (JWK) object. The JWKS usually contains only one key, but it can contain two keys during a key rotation overlap period.

The JWK for an elliptic curve public key based on the P-256 curve must include the following parameters:

#### Parameter    Definition

---

|                  |   |
|------------------|---|
| <code>kty</code> | The family of cryptographic algorithms used with the key. This must be <code>EC</code> .  |
| <code>kid</code> | A unique identifier for a specific key within the set. This value will be referenced in the status list JWT header to show which key must be used for verification. This parameter is important for associating the correct public key with the status list being verified. |
| <code>crv</code> | The cryptographic curve used with the key. This must be <code>P-256</code> .  |
| <code>x</code>   | The “x” coordinate for the elliptic curve point.  |
| <code>y</code>   | The “y” coordinate for the elliptic curve point.  |
| <code>alg</code> | The cryptographic algorithm used with the key. This must be <code>ES256</code> .  |
| <code>use</code> | The intended use of the key. This must be <code>sig</code> to indicate the key can be used to verify the signature.   |

---

### JWKS example

Below is an example of a JWKS containing one elliptic curve public key based on the P-256 curve:

```
{  
  "keys": [  
    {  
      "kty": "EC",  
      "use": "sig",  
      "crv": "P-256",  
      "kid": "5dcbee863b5d7cc30c9ba1f7393dacc6c16610782e4b6a191f94a7e8b1e151",  
      "x": "6jCKX_QRrmTeEJi-uiwcYqu8BgMgl70g2pdAst24MPE",  
      "y": "icPzjbSk6apD_SNvQt8NW0P1PeGG4KYU55GfnARryoY",  
      "alg": "ES256"  
    }  
  ]  
}
```

This page was last reviewed on 22 October 2025. It needs to be reviewed again on 22 April 2026 .



## Accessibility

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Use Status List Service to change credential status

The Status List Service keeps a record of the status of all credentials issued to GOV.UK Wallet that include status list functionality.

The Status List Service provides:

- an issuer-facing API that credential issuers can use to issue a status list entry to a credential, and revoke it when required
- a set of public-facing credential status lists that anyone can use to check a credential's status

As a credential issuer, you can use [the issuer-facing API](#) to request a unique `uri` and `index` which can be embedded into a new credential when you issue it to GOV.UK Wallet. The `uri` and `index` refer to a unique slot in one of the Status List Service's status lists, where your credential's status will be stored. You can then [set the status of this issued status list slot as `revoked`](#) if you need to revoke a credential in future.

Anyone, including credential verifiers and holders, can use the [public-facing status lists](#) to get a single status list and check the status of a credential it contains.

In this section you can find information about:

- what to do [before you issue a status record](#)
- [issuing a status record](#)
- [revoking a credential](#)
- [checking a credential's status](#)

This page was last reviewed on 22 October 2025. It needs to be reviewed again on 22 April 2026 .



## Accessibility

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Use sample reference material

You can use our test harness and sample reference data to help you issue credentials to GOV.UK Wallet correctly.

## Test harness

You can use the [GOV.UK Wallet test harness](https://github.com/govuk-one-login/mobile-wallet-cri-test-harness) (<https://github.com/govuk-one-login/mobile-wallet-cri-test-harness>) to validate your credential issuance implementation.

Follow the instructions in the README files to run the test scripts on your credential issuer.

## Example credential issuer

You can use the [credential issuer service in Java](https://github.com/govuk-one-login/mobile-wallet-example-credential-issuer) (<https://github.com/govuk-one-login/mobile-wallet-example-credential-issuer>) to see an example of a credential issuer service integrated with GOV.UK Wallet.

Follow the instructions in the README files to run an example of the credential issuance flow.

 **You should not use the sample reference material in a production environment.**

This page was last reviewed on 15 August 2025. It needs to be reviewed again on 15 February 2026 .



## Accessibility

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# GOV.UK Wallet

GOV.UK Wallet lets users save and share digital versions of their government documents on their smartphone or device.

## Issuing a document as a government service

If you work in a central government department, you can use GOV.UK Wallet to issue digital and verifiable versions of the physical documents you issue already as part of your service.

Before you can issue credentials to GOV.UK Wallet, you must already be using [GOV.UK One Login](#) (<https://www.sign-in.service.gov.uk/>) in your service to log in your users or verify their identities. You should discuss using GOV.UK Wallet with your GOV.UK One Login Engagement Manager, or [contact us](#) (/contact-us.html).

This documentation helps people such as developers and product managers of services that are using GOV.UK One Login to:

- learn how to issue digitally verifiable credentials to GOV.UK Wallet as part of their service journey
- understand [how GOV.UK Wallet works for credential issuers](#) (/issuing-credentials-to-wallet.html)
- [prepare your service to use GOV.UK Wallet](#) (/before-integrating.html)
- understand [how organisations and individuals can receive and validate digital credentials you have issued](#) when presented by the holder
- access [sample reference material](#) (/use-sample-reference-material.html) to validate your implementation

## Using and consuming a document

You will be able to consume credentials from GOV.UK Wallet if you are:

- a public sector organisation (such as central government, the NHS, or a local authority)
- a certified Digital Verification Services (DVS) provider on the [digital identity and attribute services register](#) (<https://www.digital-identity-services-register.service.gov.uk/>)

This documentation helps developers and service teams in these organisations to:

- understand [how GOV.UK wallet works for consuming credentials](#)
- understand the [open standards that will allow their service to consume and verify credentials and attributes \(/consuming-and-verifying-credentials/supported-protocols.html\)](#)

This documentation will be updated as new information and features become available.

Digital verification services should read the guidance about [using GOV.UK Wallet in the digital identity sector \(https://gov.uk/guidance/using-govuk-wallet-in-the-digital-identity-sector\)](#) to understand the different models available.

## Documentation updates

These are the most recent changes to the GOV.UK Wallet technical documentation.

| Publication date | Update  |
|------------------|---|
| Oct 22 2025      | Add new ' <a href="#">Use Status List Service to change credential status</a> ' section to add guidance on using the Status List Service. Includes information about the <a href="#">/issue</a> and <a href="#">/revoke</a> APIs, and guidance on <a href="#">checking a credential's status</a> .  |
| Aug 15 2025      | Add new ' <a href="#">Use sample reference material (/use-sample-reference-material.html)</a> ' page to link to reference implementation and test harness resources.  |
| Jul 10 2025      | Add <code>exp</code> as a required claim in the GOV.UK One Login access token.  |
| Jul 10 2025      | Clarify requirements for date fields in JWT VC credentials: Remove Optional claims <code>exp</code> and <code>nbf</code> . Update guidance on credential expiration. Specify <code>validFrom</code> and <code>validUntil</code> must be expressed in ISO 8601 format with seconds. Update <code>validUntil</code> to be a Required claim. |
| Jul 03 2025      | Remove <code>credentialSubject</code> , <code>display</code> and <code>key_attestations_required</code> parameters, and rename <code>cryptographic_suites_supported</code> to <code>credential_signing_alg_values_supported</code> in Metadata API.   |
| Jul 02 2025      | Update the second value for <code>@context</code> property in DID Document.   |
| Jun 26 2025      | Update guidance on credential photos.   |

---

|             |   |
|-------------|---|
| Jun 19 2025 | Update proof of possession <code>iat</code> claim description. Add <code>Cache-Control</code> with value of <code>no-store</code> to Notification and Credential APIs. Add 400 Bad Request response to Credential API.  |
| May 29 2025 | Add a 401 Unauthorized response to the Credential API.  |
| May 28 2025 | Update the Notification API 401 Unauthorized response format to comply with RFC 6750 (Bearer Token Usage).  |
| May 14 2025 | Add <a href="#">Consuming and verifying credentials</a> section for credential consumer and verifier documentation. Update <a href="#">GOV.UK Wallet (/)</a> page with new introductory content for new section.  |
| Apr 23 2025 | Restructure ‘What GOV.UK Wallet does, and how it works’ into a <a href="#">‘GOV.UK Wallet (/)’</a> introductory page and a <a href="#">‘How GOV.UK Wallet works (/issuing-credentials-to-wallet.html)’</a> page. Remove ‘Understand GOV.UK Wallet’s credential exchange flow’ page, and move issuance flow diagram and steps into the new <a href="#">‘How GOV.UK Wallet works (/issuing-credentials-to-wallet.html)’</a> page. |
| Apr 16 2025 | Begin recording changes in a changelog.   |

---

This page was last reviewed on 14 May 2025. It needs to be reviewed again on 14 November 2025 .



## Accessibility

## **OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

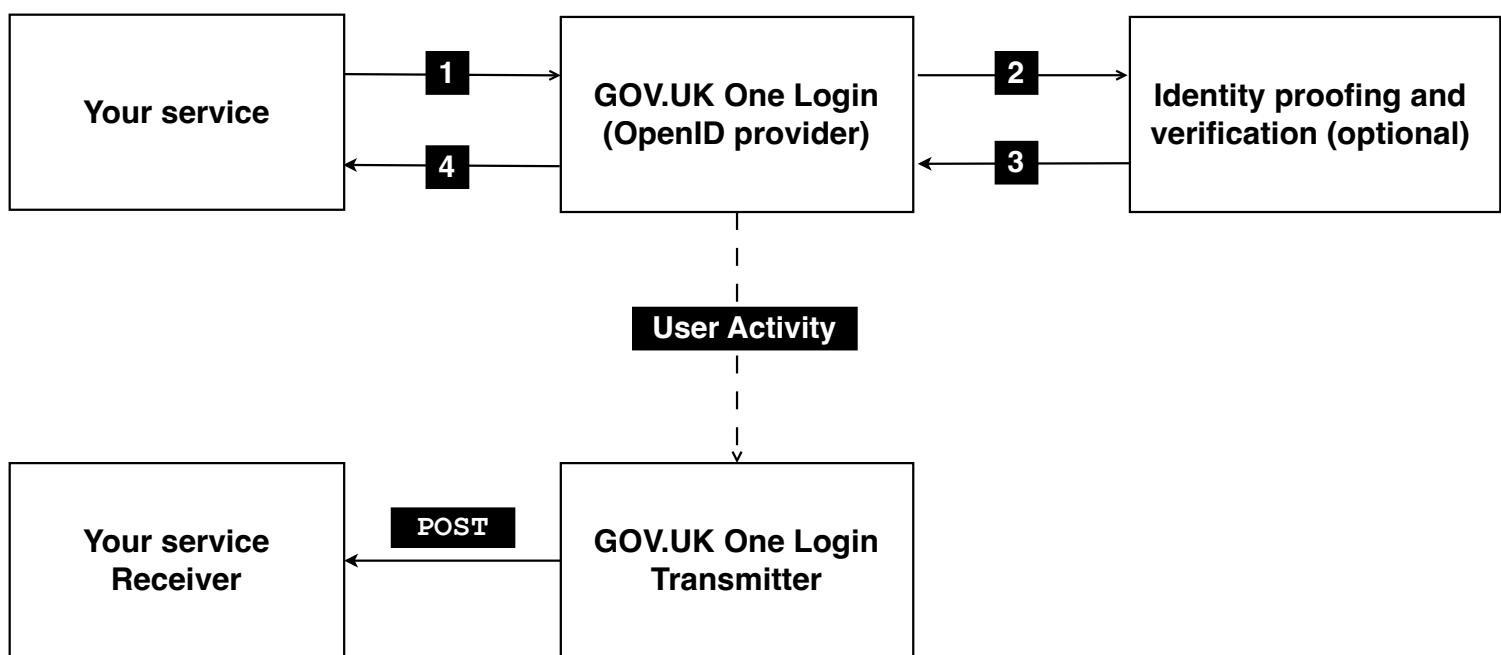
[© Crown copyright](#)

[Table of contents](#)

# How the GOV.UK One Login Signal Exchange works

When a user is redirected from your service to GOV.UK One Login, One Login records the user's activity during their interaction. As the user interacts with One Login, certain activities will trigger different signals which the GOV.UK One Login Signal Transmitter may send programmatically to your receiver using a `POST` request.

The signals will be sent asynchronously and independently of the One Login flow.



## What kind of signals can One Login send to you

The GOV.UK One Login Signal Exchange uses the [Shared Signals Events Framework](https://openid.github.io/sharedsignals/openid-sharedsignals-framework-1_0.html) ([https://openid.github.io/sharedsignals/openid-sharedsignals-framework-1\\_0.html](https://openid.github.io/sharedsignals/openid-sharedsignals-framework-1_0.html)) (SSE) to send user related activity. Signals that are supported are:

- Specific to users interacting with GOV.UK One Login with your service.
- Signals related to the user but not to an interaction with your service.



**You cannot receive signals about users interacting with GOV.UK One Login with different government services**

Find out what to consider before you [integrate your service with the GOV.UK One Login Signal Exchange \(/govuk-one-login-signal-exchange/before-you-integrate/\)](#).

This page was last reviewed on 19 February 2025.

[View source](#) [Report problem](#) [GitHub Repo](#)



[Accessibility](#)

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Before you integrate with the GOV.UK One Login Signal Exchange

Before you integrate with the GOV.UK One Login Signals Exchange, you must decide on:

- A value to be used as part of the [audience](#) (<https://datatracker.ietf.org/doc/html/rfc7519#section-4.1.3>) claim in the [Security Event Token](#) (<https://www.rfc-editor.org/info/rfc8417>) (SET). For example:  
<https://notification.department.gov.uk>
- What signal(s) you would like to receive - the GOV.UK One Login Signal Exchange team can help you with this.
- Your [expected throughput](#) (</govuk-one-login-signal-exchange/integrating-with-govuk-one-login-signal-exchange/receiver/#signal-delivery>) of users to use GOV.UK One Login.
- A support process in case we need to contact you.

As a minimum, you should be able to provide a value for the [audience](#) claim so that we can provide a [clientId](#) and [clientSecret](#) that you can use for authenticating against the GOV.UK One Login Signals Exchange Transmitter API.

You must also set up:

- An [OAuth2 service](#) (</govuk-one-login-signal-exchange/integrating-with-govuk-one-login-signal-exchange/oauth/>) that supports the [OAuth2 Client Credential Grant](#) (<https://oauth.net/2/grant-types/client-credentials/>).
- A [receiver service](#) (</govuk-one-login-signal-exchange/integrating-with-govuk-one-login-signal-exchange/receiver/>) that can receive signals sent by the GOV.UK One Login Transmitter.

You should also:

- Set up a health check using the [verification endpoint](#) (</govuk-one-login-signal-exchange/integrating-with-govuk-one-login-signal-exchange/health/>).

You can:

- [View your configuration \(/govuk-one-login-signal-exchange/integrating-with-govuk-one-login-signal-exchange/configuration/\).](#)

Find out what you need to do to start [developing your integration \(/govuk-one-login-signal-exchange/integrating-with-govuk-one-login-signal-exchange/\)](#).

This page was last reviewed on 19 February 2025.

[View source](#) [Report problem](#) [GitHub Repo](#)



## Accessibility

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Configuring for production

Configuring for production is similar to the [integration environment \(/govuk-one-login-signal-exchange/testing-your-integration/\)](#). First getting in touch with your Engagement manager to say that you're ready to integrate in production. We will then perform a health check to establish a connectivity test our production environment and your production receiver.

We will perform then perform some additional changes to the GOV.UK One Login Signal Exchange to enable the delivery of the signals that have been agreed upon.

Once all the changes are made, any user that goes through your production environment will generate signals and these will be delivered to your receiver.

Once the production configuration is complete, you can continue with [configuring your service for production \(/configure-for-production/\)](#)

This page was last reviewed on 19 February 2025.

[View source](#) [Report problem](#) [GitHub Repo](#)



## Accessibility

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# View Your configuration

The GOV.UK One Login Signals Exchange Transmitter API uses the [Stream Endpoint](https://openid.github.io/sharedsignals/openid-sharedsignals-framework-1_0.html#name-stream-configuration) ([https://openid.github.io/sharedsignals/openid-sharedsignals-framework-1\\_0.html#name-stream-configuration](https://openid.github.io/sharedsignals/openid-sharedsignals-framework-1_0.html#name-stream-configuration)) as a way to view the configuration of the integration.

In order to view your configuration, you must first authenticate against the GOV.UK One Login Transmitter OAuth token endpoint:

```
curl --request POST \
--url TOKEN ENDPOINT \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data client_id=CLIENT_ID \
--data client_secret=CLIENT_SECRET \
--data grant_type=client_credentials
```

Where:

- **CLIENT\_ID** : The `clientId` supplied to you [before you integrate \(/govuk-one-login-signal-exchange/before-you-integrate\)](#).
- **CLIENT\_SECRET** : The `clientSecret` supplied to you [before you integrate \(/govuk-one-login-signal-exchange/before-you-integrate\)](#).
- **ENDPOINT** : The token endpoint of Transmitter API. Below are the token endpoints for our
  - [Integration Environment](https://auth.ssf-transmitter.transaction.integration.account.gov.uk/oauth2/token) (<https://auth.ssf-transmitter.transaction.integration.account.gov.uk/oauth2/token>).
  - [Production Environment](https://auth.shared-signals-transmitter.transaction.account.gov.uk/oauth2/token) (<https://auth.shared-signals-transmitter.transaction.account.gov.uk/oauth2/token>).

The expected response should look like the following:

```
{
  "access_token": "ACCESS_TOKEN",
```

```
"token_type": "bearer",  
"expires_in": 14400  
}
```

Once you've obtained an **ACCESS\_TOKEN**, you can make the following request to view your configuration:

```
curl --request GET \  
--url STREAM_ENDPOINT \  
--header 'Authorization: Bearer ACCESS_TOKEN' \  
--header 'Content-Type: application/json' \  

```

Below are the stream endpoints for our:

- [Integration \(`https://ssf-transmitter.transaction.integration.account.gov.uk/stream`\)](https://ssf-transmitter.transaction.integration.account.gov.uk/stream) Environment.
- [Production \(`https://shared-signals-transmitter.transaction.account.gov.uk/stream`\)](https://shared-signals-transmitter.transaction.account.gov.uk/stream) Environment.

Once you have managed to view your configuration, you can then begin testing in the GOV.UK One Login [integration environment \(/govuk-one-login-signal-exchange/testing-your-integration/\)](/govuk-one-login-signal-exchange/testing-your-integration/). If you are unable to view your configuration or the configuration is not what you expect, please get in touch with GOV.UK One Login support team and we can investigate further.

This page was last reviewed on 19 February 2025.

[View source](#) [Report problem](#) [GitHub Repo](#)



[Accessibility](#)

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Perform a Health check

The health check uses the [Verification Endpoint](https://openid.github.io/sharedsignals/openid-sharedsignals-framework-1_0.html#name-verification) ([https://openid.github.io/sharedsignals/openid-sharedsignals-framework-1\\_0.html#name-verification](https://openid.github.io/sharedsignals/openid-sharedsignals-framework-1_0.html#name-verification)) as a way to confirm the connection between the GOV.UK One Login Transmitter and your receiver as well as a way to test your [receiver](/govuk-one-login-signal-exchange/integrating-with-govuk-one-login-signal-exchange/receiver/) (</govuk-one-login-signal-exchange/integrating-with-govuk-one-login-signal-exchange/receiver/>).

In order to send a health check, you must first authenticate against the GOV.UK One Login Transmitter OAuth token endpoint:

```
curl --request POST \
--url TOKEN ENDPOINT \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data client_id=CLIENT_ID \
--data client_secret=CLIENT_SECRET \
--data grant_type=client_credentials
```

Where:

- **CLIENT\_ID** : The `clientId` supplied to you [before you integrate](/govuk-one-login-signal-exchange/before-you-integrate/) (</govuk-one-login-signal-exchange/before-you-integrate/>).
- **CLIENT\_SECRET** : The `clientSecret` supplied to you [before you integrate](/govuk-one-login-signal-exchange/before-you-integrate/) (</govuk-one-login-signal-exchange/before-you-integrate/>).
- **ENDPOINT** : The token endpoint of Transmitter API. Below are the token endpoints for our
  - [Integration Environment](https://auth.ssf-transmitter.transaction.integration.account.gov.uk/oauth2/token) (<https://auth.ssf-transmitter.transaction.integration.account.gov.uk/oauth2/token>).
  - [Production Environment](https://auth.shared-signals-transmitter.transaction.account.gov.uk/oauth2/token) (<https://auth.shared-signals-transmitter.transaction.account.gov.uk/oauth2/token>).

The expected response should look like the following:

```
{  
  "access_token": "ACCESS_TOKEN",  
  "token_type": "bearer",  
  "expires_in": 14400  
}
```

Once you've obtained an `ACCESS_TOKEN` you can then send a `POST` request to the verification endpoint:

```
curl --request POST \  
  --url VERIFICATION_ENDPOINT \  
  --header 'Authorization: Bearer ACCESS_TOKEN' \  
  --header 'Content-Type: application/json' \  
  --data PAYLOAD
```

Where the optional `PAYLOAD` field is the following JSON object converted to a JSON string:

```
{  
  "state": "abc123"  
}
```

Below are the verification endpoints for our:

- [Integration \(`https://ssf-transmitter.transaction.integration.account.gov.uk/verify`\)](https://ssf-transmitter.transaction.integration.account.gov.uk/verify) Environment.
- [Production \(`https://shared-signals-transmitter.transaction.account.gov.uk/verify`\)](https://shared-signals-transmitter.transaction.account.gov.uk/verify) Environment.

Once the request is made, the value of `state` will be included in the verification signal to your receiver endpoint if supplied. The value of `state` must be:

- Any alphanumeric character.
- Maximum length of 64 characters.
- Only special character allowed is `-`

You should send a health check on a recurring schedule at a sensible rate, usually every 5-10 minutes. If you do not receive a verification signal or the signal does not contain the expected the state value, please get in touch with GOV.UK One Login support team and we can investigate further.

You can also [view your configuration](/govuk-one-login-signal-exchange/integrating-with-govuk-one-login-signal-exchange/configuration/) (/govuk-one-login-signal-exchange/integrating-with-govuk-one-login-signal-exchange/configuration/).

This page was last reviewed on 19 February 2025.

[View source](#) [Report problem](#) [GitHub Repo](#)



[Accessibility](#)

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# OAuth Token Endpoint

The GOV.UK One Login Signal Exchange uses the [OAuth2 Client Credentials Grant](https://oauth.net/2/grant-types/client-credentials/) (<https://oauth.net/2/grant-types/client-credentials/>) to authenticate between GOV.UK One Login Signal Exchange transmitter and your receiver when sending signals.

The GOV.UK One Login Transmitter will initiate the client credentials flow by making this API request:

```
curl --request POST \
--url https://YOUR.SERVICE.gov.uk/oauth2/token \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data client_id=CLIENT_ID \
--data client_secret=CLIENT_SECRET \
--data grant_type=client_credentials
```

The expected response should look like the following:

```
{
  "access_token": "ACCESS_TOKEN",
  "token_type": "bearer",
  "expires_in": 14400,
  "scope": "hello"
}
```

## Token Endpoint behaviour

There are certain assumptions that we make about your endpoint. These are:

- Validity of `access_token` :

- GOV.UK One Login Signal Exchange Transmitter must be able to request a new token without invalidating previously issued tokens.
  - The token must be valid for a minimum of 1 hour.
- Throughput:
    - The `/token` endpoint will need to be able to support a [high throughput \(/govuk-one-login-signal-exchange/integrating-with-govuk-one-login-signal-exchange/receiver/#signal-delivery\)](#).

The token endpoint will then be used to authenticate against the [receiver endpoint \(/govuk-one-login-signal-exchange/integrating-with-govuk-one-login-signal-exchange/receiver/\)](#) which you must set up next.

This page was last reviewed on 19 February 2025.

[View source](#) [Report problem](#) [GitHub Repo](#)



## Accessibility

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Receiver Endpoint

The receiver endpoint is where the GOV.UK One Login Signal Exchange Transmitter will send signals related to user activity and should be built as per [RFC 8935](https://www.rfc-editor.org/rfc/rfc8935.html) (<https://www.rfc-editor.org/rfc/rfc8935.html>) .

After obtaining an **ACCESS\_TOKEN** from the [OAuth Token Endpoint \(/govuk-one-login-signal-exchange/integrating-with-govuk-one-login-signal-exchange/oauth/\)](/govuk-one-login-signal-exchange/integrating-with-govuk-one-login-signal-exchange/oauth/), the transmitter will make the following request when attempting to send the signal:

```
curl --request POST \
--url https://your.service.gov.uk/receiver \
--header 'Accept: application/json' \
--header 'Authorization: Bearer ACCESS_TOKEN' \
--header 'Content-Type: application/secevent+jwt' \
--data SIGNAL
```

**SIGNAL** is a [Security Event Token](https://www.rfc-editor.org/info/rfc8417) (<https://www.rfc-editor.org/info/rfc8417>) (SET) which builds upon the [JSON Web Token \(JWT\)](https://www.rfc-editor.org/rfc/rfc7519) (<https://www.rfc-editor.org/rfc/rfc7519>) format and can be decoded as a regular JWT. You must return a **HTTP 202 (Accepted)** as any other responses will be treated as an error and will be attempted to be [delivered again](/govuk-one-login-signal-exchange/integrating-with-govuk-one-login-signal-exchange/receiver/#signal-delivery) (</govuk-one-login-signal-exchange/integrating-with-govuk-one-login-signal-exchange/receiver/#signal-delivery>).

## Signature Verification

As part of the SET, there is a signature that should be used to verify the authenticity of the payload:

1. Validate that the SET **kid** claim exists in the JWKS (JSON web key set) returned by the </jwks> endpoint.
2. Check the JWT **alg** header matches the value for the key you are using.

3. Use the key to validate the signature on the logout token according to the [JSON Web Signature Specification](https://datatracker.ietf.org/doc/html/rfc7515) (<https://datatracker.ietf.org/doc/html/rfc7515>).
4. Check the value of `iss` (issuer) matches <https://ssf.account.gov.uk/>.
5. Check the `aud` (audience) claim is the same client ID you received when you [registered your service to use GOV.UK One Login Signal Exchange](#) ([/govuk-one-login-signal-exchange/before-you-integrate/](#)).
6. Check the `iat` (issued at) claim is in the past.

Below are the JWKS endpoints for our:

- [Integration](https://ssf-transmitter.transaction.integration.account.gov.uk/jwks.json) (<https://ssf-transmitter.transaction.integration.account.gov.uk/jwks.json>) Environment.
- [Production](https://shared-signals-transmitter.transaction.account.gov.uk/jwks.json) (<https://shared-signals-transmitter.transaction.account.gov.uk/jwks.json>) Environment.

As we may send SETs at a [high throughput](#) ([/govuk-one-login-signal-exchange/integrating-with-govuk-one-login-signal-exchange/receiver/#signal-delivery](#)), you must implement caching of the JWKS endpoint. The cache must:

- Refresh regularly.
- Refresh if the value in the `kid` claim does not exist in your cache.

## Payload validation

As the SET is a JWT, the body will be the signal itself. After the body has been decoded, we recommend validating the payload. The GOV.UK One Login Signal Exchange team can provide JSON schema's to help with payload validation.

## Signal Delivery

A single user interaction with GOV.UK One Login may generate several signals therefore during the development of your receiver, you need to ensure that your solution can support a large volume of signals.

The GOV.UK One Login Signal Exchange team will work with you to identify the expected delivery rate of signals based on a number of factors, as well as implement rate limiting to prevent your infrastructure from being overwhelmed with signals. These factors may include:

- Your expected throughput of users attempting to use your service.
- The signal(s) that you have subscribed to.
- If you are strictly an Auth only service or if you are an Identity Proofing and Verification Service.

We recommend that your receiver is **initially** setup to support minimum of 10 signals per second until a more accurate number can be determined based on the factors above.

-  **You must not conduct any security testing, penetration testing, performance testing, or IT health checks of the GDS estate. You must also not use personal identifiable information (PII) – GOV.UK One Login will provide example data.**

## Transmitter Assumptions

You should also assume that the GOV.UK One Login Transmitter may deliver:

- Signals out of order.
- Duplicate signals.

GOV.UK One Login Transmitter will make a best effort attempt to ensure that signals are delivered in order and that there is no duplication.

## Transmitter Error Handling

If the transmitter receives any response other than [HTTP 202 \(Accepted\)](#), then it is considered an error. The transmitter may attempt redelivery every 2 minutes up to 5 times. After the 5th attempt, the failed signals may be kept for up to 14 days before they are discarded. The GOV.UK One Login Signal Exchange will get in touch via the support process to arrange a replay.

Once you have set up your transmitter, you may test the integration by either:

- Setting up a [health check \(/govuk-one-login-signal-exchange/integrating-with-govuk-one-login-signal-exchange/health/\)](#).
- Testing in the GOV.UK One Login [integration environment \(/govuk-one-login-signal-exchange/testing-your-integration/\)](#).

This page was last reviewed on 19 February 2025.



## Accessibility

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Integrating with One Login Signal Exchange

Before integrating with the GOV.UK One Login Signal Exchange, there are several services (</govuk-one-login-signal-exchange/before-you-integrate/>) that need to be built.

To get started, first setup an [OAuth2 Service](#) (</govuk-one-login-signal-exchange/integrating-with-govuk-one-login-signal-exchange/oauth/>).

This page was last reviewed on 19 February 2025.

[View source](#) [Report problem](#) [GitHub Repo](#)



[Accessibility](#)

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Support

Use the [#govuk-one-login channel](#)

(<https://ukgovernmentdigital.slack.com/archives/C02AQUJ6WTC>) to contact the GOV.UK One Login technical team.

This page was last reviewed on 19 February 2025.

[View source](#) [Report problem](#) [GitHub Repo](#)



[Accessibility](#)

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# Testing your integration in GOV.UK One Login integration environment

Once you have built the necessary [services \(/govuk-one-login-signal-exchange/before-you-integrate/\)](#), you can start testing your integration with GOV.UK One Login and the GOV.UK One Login Signal Exchange

To get started, first get in touch with your Engagement Manager, they will work with you and the GOV.UK One Login Signal Exchange team to perform a [health check \(/govuk-one-login-signal-exchange/integrating-with-govuk-one-login-signal-exchange/health/\)](#). By performing a health check, we are establishing connectivity between the GOV.UK One Login Signal Exchange integration environment and your receiver.

Once the health check is complete, the rest of the integration requirements will need to be [addressed \(/govuk-one-login-signal-exchange/before-you-integrate/\)](#) and some additional changes will be made by the GOV.UK One Login Signal Exchange to enable the delivery of the signals that have been agreed upon.

Once the changes are complete, you can continue with your [integration \(/integrate-with-integration-environment/\)](#) with GOV.UK One Login and begin [testing \(/test-your-integration/\)](#) your integration with GOV.UK One Login as normal.

As you interact with One Login's integration environment, signals will be generated and delivered to your receiver.

After testing is complete, you can move towards [Configuring your service for Production \(/govuk-one-login-signal-exchange/configuring-for-production/\)](#)

 **You must not conduct any security testing, penetration testing, performance testing, or IT health checks of the GDS estate. You must also not use personal identifiable information (PII) – GOV.UK One Login will provide example data.**



## Accessibility

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)

[Table of contents](#)

# About GOV.UK One Login Signal Exchange

The GOV.UK One Login Signal Exchange is a way for both GOV.UK One Login and relying parties to share signals related to the users who are using their systems. This technical documentation gives you information on:

- What the GOV.UK One Login Signal Exchange is
- How it works
- How to configure your service to use the service

To get started, contact your Engagement Manager – if you do not have an Engagement Manager, [complete the form to register your interest \(<https://www.sign-in.service.gov.uk/register>\)](https://www.sign-in.service.gov.uk/register).

Learn [How the GOV.UK One Login Signal Exchange works \(</govuk-one-login-signal-exchange/about-govuk-one-login-signal-exchange-works/>\)](/govuk-one-login-signal-exchange/about-govuk-one-login-signal-exchange-works/)

## Documentation updates

These are the most recent changes to this documentation.

| Publication date | Update  |
|------------------|---|
| February 19 2025 | Published initial version of the GOV.UK One Login Signal Exchange documentation |

This page was last reviewed on 19 February 2025.

[View source](#) [Report problem](#) [GitHub Repo](#)



[Accessibility](#)

**OGL**

All content is available under the [Open Government Licence v3.0](#), except where otherwise stated

[© Crown copyright](#)