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

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/ (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/) 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) 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

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).

GOV.UK One Login uses 'Vectors of Trust' (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	`Cl` (credential low)	This vector requires your users to have a username and password combination. 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. All services use `Cl.Cm` as the authentication level by default, unless you change your authentication level to `Cl` . If you request `Cl` , you will not be able to request identity attributes.
Medium level of protection	`Cl.Cm` (credential medium)	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. All services use `Cl.Cm` as the authentication level by

default, unless you change your authentication level to `Cl` .

If you need to request identity attributes, you must request `Cl.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.

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). 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 imposters or fake identities with fabricated credentials, otherwise known as 'synthetic identities'. To request a medium level of identity confidence (`P2`), you must have specified `Cl.Cm` (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

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 request user attributes using scopes (/before-integrating/choose-which-user-attributes-your-service-can-request/#choose-which-scopes-your-service-can-request).
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	OIDC requests to the `/authorize` endpoint must contain the `openid` scope value to indicate that an application intends to use the OIDC protocol. This will return the sub claim, which uniquely identifies your user.
`email`	Optional	Returns the `email` claim, which contains: • your user's email address • `email_verified`, which is a boolean indicating whether your user has verified their email address or not
`phone`	Optional	Returns the `phone_number` claim, which contains: • your user's phone number • `phone_number_verified`, which is a boolean indicating whether your user has verified their phone number or not
`wallet-subject-id`	Optional (required to use GOV.UK Wallet)	Returns the `walletSubjectId` 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. 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. The value is returned in the format: `urn:fdc:wallet.account.gov.uk:2024:3c_jJtXcLttICSNrkW7M1vO2_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.

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
`https://vocab.account.gov.uk/v1/coreIdentityJWT`	This claim contains core identity information about your user: • their names • their date of birth • the level of confidence GOV.UK One Login reached in your user's identity
`https://vocab.account.gov.uk/v1/address`	This claim contains your user's postal addresses.
`https://vocab.account.gov.uk/v1/passport`	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.
`https://vocab.account.gov.uk/v1/drivingPermit`	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.
`https://vocab.account.gov.uk/v1/returnCode`	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": "C"}]` , in the response. For security reasons, you'll have to contact GOV.UK One Login on govuk-one-login@digital.cabinet-office.gov.uk 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 `access_denied` error instead. There's further guidance on the `returnCode` claim (/integrate-with-integration-environment/prove-users-identity/#understand-your-user-s-return-code-claim).

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.

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 `https://mythical-creatures.gov.uk` as the `sector_identifier_uri` for all 3 services.
Prevent services from sharing users	Set the sector identifier in each service to a different value.	Set a separate `sector_identifier_uri` for each service: • `https://register-your-hydra.mythical-creatures.gov.uk` • `https://tax-your-dragon.mythical-creatures.gov.uk` • `https://report-a-unicorn.mythical-creatures.gov.uk`
Share users across some services	Set the sector identifier in the services that share users to the same value. Give the services that should not share users a different sector identifier.	Set `https://mythical-creatures.gov.uk` as the `sector_identifier_uri` for 'register your hydra' and 'report a unicorn' to share users. Set `https://tax-your-dragon.mythical-creatures.gov.uk` as the `sector_identifier_uri` for 'tax your dragon' to have a separate user base.

This page was last reviewed on 9 February 2024.

Accessibility

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) 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.

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 (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	There's guidance on configuring Amazon Cognito to use GOV.UK One Login as an external OpenID Connect provider (https://github.com/govuk-one-login/onboarding-examples/tree/main/clients/amazon-cognito) (opens separate repository).
Salesforce	You'll need to build an authentication provider plugin to integrate using Salesforce. There's further guidance on building an authentication provider plugin (https://github.com/govuk-one-login/onboarding-examples/blob/main/clients/salesforce-apex/README.md) (opens separate repository).

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 (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/).
2. Run `brew install openssl`.

For Windows:

1. Follow the documentation to install Chocolatey (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/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 (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.

Accessibility

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 guidance on naming your service (https://www.gov.uk/service-manual/design/naming-your-service).
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 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).
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 requesting logout notifications from GOV.UK One Login (/integrate-with-integration-environment/managing-your-users-sessions/#request-logout-notifications-from-gov-uk-one-login).
Landing Page URL	It's not possible to configure this yet. Send an email to govuk-one-login@digital.cabinet-office.gov.uk 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)].

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 `private_key_jwt` or `client_secret_post`. There's further guidance on using the correct token authentication method for your service (/before-integrating/use-correct-token-authentication-method/).
Public key	Only include this if your service is using the `private_key_jwt` 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 guidance on generating a key pair (/before-integrating/set-up-your-public-and-private-keys/#create-a-key-pair).
ID token signing algorithm	Choose either `RS256` or `ES256`. By default, GOV.UK One Login will sign the `id_token` JSON Web Token (JWT) using the `ES256` algorithm but some third party tooling does not support `ES256`. If your service needs an alternative algorithm, we can sign your `id_token` JWT using the `RS256` algorithm

This page was last reviewed on 25 September 2024.

Accessibility

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": "pSx43eUV2hZ3AJKYNFHxOsILQ_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.

Accessibility

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) 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.

Accessibility

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

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

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

`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 `http://` and `localhost`. There's further guidance on requesting logout notifications from GOV.UK One Login (/integrate-with-integration-environment/managing-your-users-sessions/#request-logout-notifications-from-gov-uk-one-login).
`ClientID`	GOV.UK One Login will fill in `ClientID` 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: • `https://vocab.account.gov.uk/v1/passport` • `https://vocab.account.gov.uk/v1/drivingPermit` • `https://vocab.account.gov.uk/v1/coreIdentityJWT` • `https://vocab.account.gov.uk/v1/address` • `https://vocab.account.gov.uk/v1/returnCode`
`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 guidance on naming your service (https://www.gov.uk/service-manual/design/naming-your-service).
`ClientType`	Leave this field as `web`.
`ConsentRequired`	Leave this field as `false`.
`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 `false`.

`IdentityVerificationSupported`	If you're using identity verification, this should be `true`. If you only need authentication, this should be `false`.
`IdTokenSigningAlgorithm`	This will be `ES256` or `RS256`. You can find the one you're using in your application's code.
`LandingPageUrl`	`LandingPageUrl` is only required if you're making identity requests. GOV.UK One Login supports a single `LandingPageUrl` 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. These must be production-grade URLs without reference to `http://` and `localhost`.
`OneLoginService`	Leave this field as `false`.
`PostLogoutRedirectUrIs`	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. These must be production-grade URLs without reference to `http://` and `localhost`. There's further guidance on 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).
`PublicKey`	`PublicKey` is only required if you're using the `private_key_jwt` 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 guidance on generating a key pair (/before-integrating/set-up-your-public-and-private-keys/#create-a-key-pair).
`IsInternalServ`	Leave this field as `false`.

`ice`	
`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 `http://` and `localhost`.
`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).
`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 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).
`ServiceType`	Leave this field as `MANDATORY`.
`SubjectType`	Leave this field as `pairwise`.
`TestClient`	Leave this field as `false`.
`TokenAuthMethod`	Specify the token authentication method your service is using. This will be `private_key_jwt` or `client_secret_post`.

	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

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
`unauthorized_client`	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 `unauthorized_client` will be returned with the error description `client deactivated`. 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.
`request_is_missing_parameters`	The request has one or more of the following issues: • missing a required parameter • includes an invalid parameter value • includes a parameter more than once • not in the correct format . 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`	The request has one or more of the following issues: • missing a required parameter • includes an invalid parameter value • includes a parameter more than once • not in the correct format . 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-Requestvtr 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 `Cl.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

Error messages from the `/userinfo` endpoint

Error	More information about your error
`invalid_token`	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 bearer token (https://oauth.net/2/bearer-tokens/).

Error messages from the `/token` endpoint

Error	More information about your error
`invalid_request`	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.
`invalid_client`	Client authentication failed, which could be caused by the request containing an invalid `client_id` or an issue in validating the signature of the `client_assertion`. 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.

This page was last reviewed on 21 January 2025.

Accessibility

How GOV.UK One Login works

GOV.UK One Login is an OpenID Connect (OIDC) (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).

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

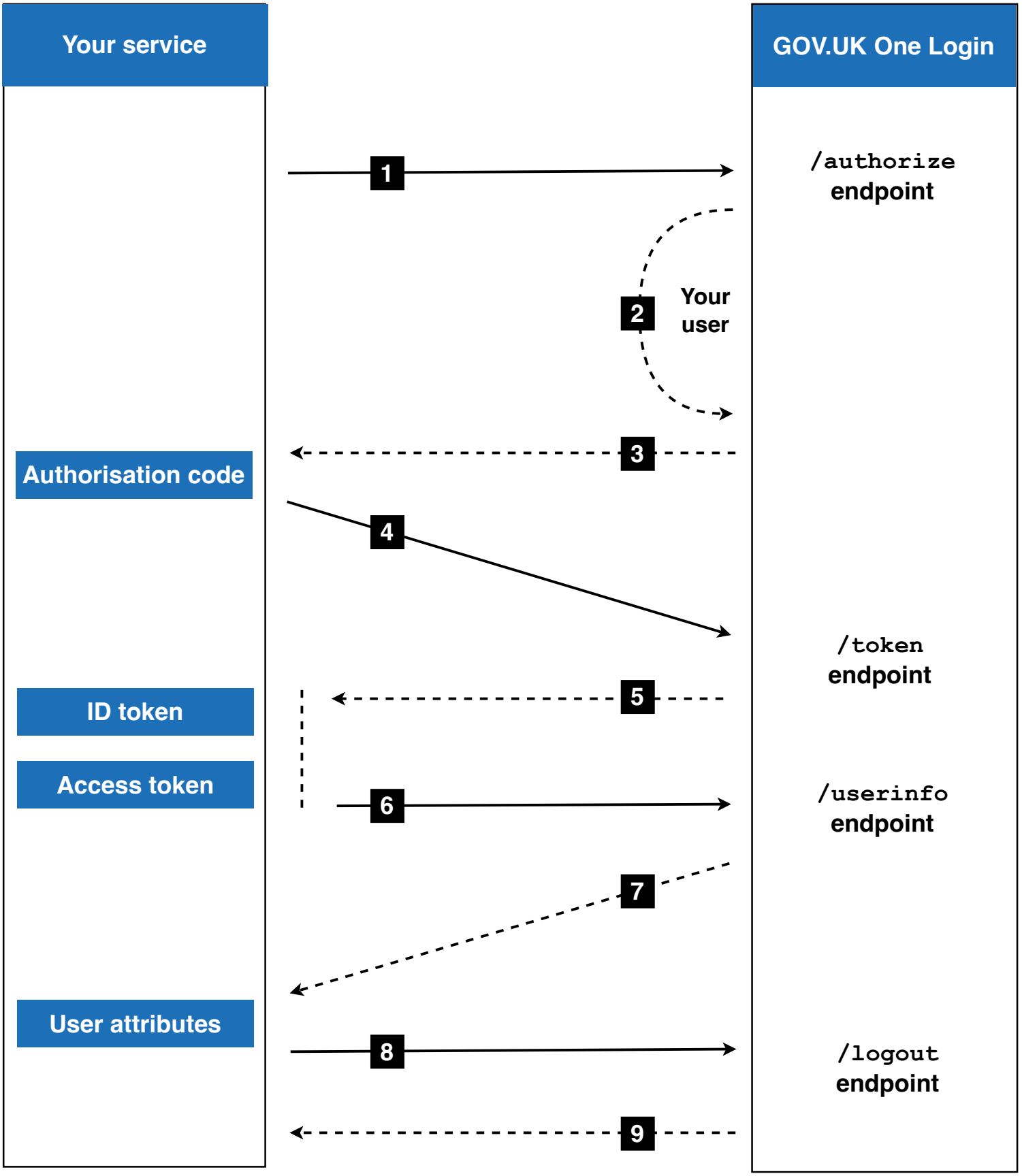
Understand the flow GOV.UK One Login uses

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) 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.

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.

Accessibility

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).

Use the integration discovery endpoint

You can use the integration discovery endpoint (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=["Cl.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 `Cl.Cm` .
2. Send a request for identity to the `/authorize` endpoint specifying the `vtr` as `Cl.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 (`Cl.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=["Cl.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).

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	You must set this value to be code: `response_type=code` . If you're using JAR, make sure the `response_type` values in the query parameters and the request object are identical.
`scope`	Required	A space-separated list of scopes. You must include `openid` as one scope value. If you request `openid` but also request other incorrect scopes, the error `invalid_scope` will return with an HTTP 302 instead. You should refer to the guidance on choosing which user attributes your service can request (/before-integrating/choose-which-user-attributes-your-service-can-request/) for the `scope` parameter. If you're using JAR, make sure the `scope` values in the query parameters and the request object are identical.
`client_id`	Required	The client identifier (/before-integrating/create-individual-configurations-for-each-service/#understanding-the-client-identifier), which we generated for you when you registered your service to use GOV.UK One Login (/before-integrating/register-and-manage-your-service/) must match your client configuration. If you're using JAR, make sure the `client_id` values in the query parameters and the request object are identical.
`state`	Required	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 `state` value solves this issue by binding the request and response, which reduces impact of Cross Site Request Forgery (https://owasp.org/www-community/attacks/csrf) attacks. This value will be returned to the client in the authentication response.
`redirect_uri`	Required	You'll have specified your `redirect_uri` when you registered your service to use GOV.UK One Login (/before-integrating/register-and-manage-your-service/). To avoid an `HTTP 400 Bad Response` error, the redirect URI must exactly match one of the URIs configured in your client configuration and also include the protocol `https://` or `http` .

		If you're using request parameters, the value must be URL-encoded.
`nonce`	Required	A unique value generated by your application that is used to verify the integrity of the `id_token` and mitigate replay attacks. This value will be present in the `id_token` and should include the per-session state, as well as being impossible for attackers to guess. Your application will need to verify the `nonce` claim value is the same as the `nonce` parameter sent in the authentication request.
`aud`	Optional	If you're using JAR, you must include `aud` in your JSON object. You must set this value to specify GOV.UK One Login's authorisation server as the intended audience: `aud=https://oidc.integration.account.gov.uk/authorize` .
`iss`	Optional	If you're using JAR, the `iss` parameter is required. You must set this value to be your `client_id` . GOV.UK One Login generated your `client_id` when you registered your service to use GOV.UK One Login (/before-integrating/register-and-manage-your-service/).
`ui_locales`	Optional	GOV.UK One Login supports English and Welsh as language choices. 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 `ui_locales` to do this. In the `ui_locales` parameter, you can choose either `en` (English) or `cy` (Welsh). Using `ui_locales` is optional. If you do not include it, your service will continue using English by default. GOV.UK One Login does not support any other languages.
`vtr`	Optional	The `vtr` 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 `vtr=["Cl.Cm.P2"]` .

		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 `Cl.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 (`Cl.Cm`). This means you will not receive identity attributes in your response.
`claims`	Optional	To get the identity attributes your service needs, you should specify these in the `claims` parameter using the `/userinfo` endpoint. The `/userinfo` endpoint returns a JSON object listing the requested claims. You can read more about choosing which user attributes your service can request (/before-integrating/choose-which-user-attributes-your-service-can-request/). You can read more about the structure of the claims request in OpenID Connect section 5.5 (https://openid.net/specs/openid-connect-core-1_0.html#ClaimsParameter).
`max_age`	Optional	`max_age` is only available to services not on the GOV.UK domain and those handling particularly sensitive data. When the `max_age` parameter is included in your request, your user will be forced to re-authenticate if the time in seconds since authentication is greater than `max_age` . `max_age` must be set to zero or a positive integer. You'll need to contact GOV.UK One Login support (https://www.sign-in.service.gov.uk/support) to request to use `max_age` .
`code_challenge`	Optional	`code_challenge` 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 `code_challenge` by transforming a `code_verifier` using a `code_challenge_method` . 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_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
`unauthorized_client`	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 `unauthorized_client` will be returned with the error description `client deactivated`. 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.
`request_is_missing_parameters`	The request has one or more of the following issues: • missing a required parameter • includes an invalid parameter value • includes a parameter more than once • not in the correct format . 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`	The request has one or more of the following issues: • missing a required parameter • includes an invalid parameter value • includes a parameter more than once • not in the correct format

	. 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-RequestvtrnotvalidRequestvtr 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 `Cl.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
`aud`	Required	`aud` stands for 'audience'. This identifies GOV.UK One Login's authorisation server as an intended audience. This value should be the URL: https://oidc.integration.account.gov.uk/token.
`iss`	Required	`iss` stands for 'issuer'. This claim should contain your `client_id` you got when you registered your service to use GOV.UK One Login (/before-integrating/register-and-manage-your-service/).
`sub`	Required	`sub` stands for 'subject'. This claim should contain your `client_id` you got when you registered your service to use GOV.UK One Login (/before-integrating/register-and-manage-your-service/). There's further guidance on how to use this value in the response to the `/userinfo` endpoint (https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/authenticate-your-user/#receive-response-for-retrieve-user-information).
`exp`	Required	`exp` 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 Unix Epoch (https://www.epochconverter.com/). 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.
`jti`	Required	`jti` 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.
`iat`	Recommended	`iat` 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 Unix Epoch (https://www.epochconverter.com/).

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=SplxlOBeZQQYbYS6WxSbIA
&redirect_uri=https%3A%2F%2Fclient.example.org%2F
&client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-type%3
&client_assertion=eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIiLCJpc3MiOiIiLCJhdWQi
OiIiLCJqdGkiOiIifQ.r1Ylfhhy6VNSlhlhW1N89F3WfIGuko2rvSRWO4yK1BI
```

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 `authorization_code` .
`redirect_uri`	Required	You'll have specified your `redirect_uri` 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 `POST` request to the `/token` endpoint.
`client_assertion_type`	Required	When you're using `private_key_jwt` , you must set the value to `urn:ietf:params:oauth:client-assertion-type:jwt-bearer` .
`code`	Required	The code you received when you generated an authorisation code (/integrate-with-integration-environment/authenticate-your-user/#generate-an-authorisation-code).
`code_verifier`	Optional	You should only include this parameter if your original `/authorize` 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": "SlAV32hkKG",
  "token_type": "Bearer",
  "expires_in": 180,
  "id_token":"eyJhbGciOiJSUzI1NiIsImtpZCI6IjFlOWdkazcifQ.ewogImlzc
    yI6ICJodHRwOi8vc2VydmVyLmV4YW1wbGUuY29tIiwKICJzdWIiOiAiMjQ4Mjg"
}
```

You can use the following table to understand the response for ‘Make a token request’.

Parameter	Description
`access_token`	The access token value is an opaque access token which you can use with the `/userinfo` endpoint to return a user’s profile.
`token_type`	The token type value. GOV.UK One Login only supports the bearer token (https://oauth.net/2/bearer-tokens/).
`expires_in`	The length of time the token is valid for. This is displayed in seconds.
`id_token`	A signed JWT that contains basic attributes about the user. By default, GOV.UK One Login signs this JWT using the `ES256` algorithm. If your service cannot support the `ES256` algorithm (for example, some third-party tooling does not support `ES256`), GOV.UK One Login can sign the JWT using the `RS256` algorithm. You’ll have specified whether your service can support `ES256` when you registered your service to use GOV.UK One Login (/before-integrating/register-and-manage-your-service/). The public key used to verify this JWT is available from the `jwks_uri`

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:VtcZjnU4Sif2oyJZola3OkN0e3Jeku1cIMN38rFlhU4",
  "aud": "YOUR_CLIENT_ID",
  "iss": "https://oidc.integration.account.gov.uk/",
  "vot": "Cl.Cm",
  "exp": 1704894526,
  "iat": 1704894406,
  "nonce": "lZk16Vmu8-h7r8L8bFFiHJxpC3L73UBpfb68WC1Qoqg",
  "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
`at_hash`	`at_hash` stands for 'access token hash'. You use `at_hash` to validate your access token. This is not mandatory. There is further guidance on `at_hash` in the Open ID Connect specification (https://openid.net/specs/openid-connect-core-1_0.html#CodeIDToken).
`sub`	`sub` stands for the subject identifier or the unique ID of a user.
`aud`	`aud` stands for the audience, which will be the `client_id` you received when you registered your service to use GOV.UK One Login (/before-integrating/register-and-manage-your-service/).
`iss`	`iss` stands for the GOV.UK One Login OpenID Provider's Issue identifier as specified in the discovery endpoint (https://oidc.integration.account.gov.uk/.well-known/openid-configuration).

`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 Unix Epoch (https://www.epochconverter.com/).
`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 Unix Epoch (https://www.epochconverter.com/).
`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 Unix Epoch (https://www.epochconverter.com/).

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).
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). 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).

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 `Cl.Cm.P2`, you must ensure the `vot` claim is equal to `Cl.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
`invalid_request`	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.
`invalid_client`	Client authentication failed, which could be caused by the request containing an invalid `client_id` or an issue in validating the signature of the `client_assertion`.

	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_02YOlWpd8PAOI-2sVlB2nsNU7mcLZYhYw=",
  "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
`sub`	The subject identifier (`sub`) 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. Do not use the `sub` as the primary identifier for your user. Instead, generate your own unique value for your user within your service and map this against the GOV.UK One Login `sub` . Mapping the `sub` makes account recovery easier. For example, if a user deletes their GOV.UK One Login, you can re-map the user’s new `sub` to your service without creating a new primary identifier for your user.

`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_jJtXcLttICSNrkW7M1vO2_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
`invalid_token`	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 bearer token (https://oauth.net/2/bearer-tokens/).

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

Helping your users after their in person identity checks

If you need identity verification, we strongly recommended that you create a landing page / `LandingPageUrl` 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.

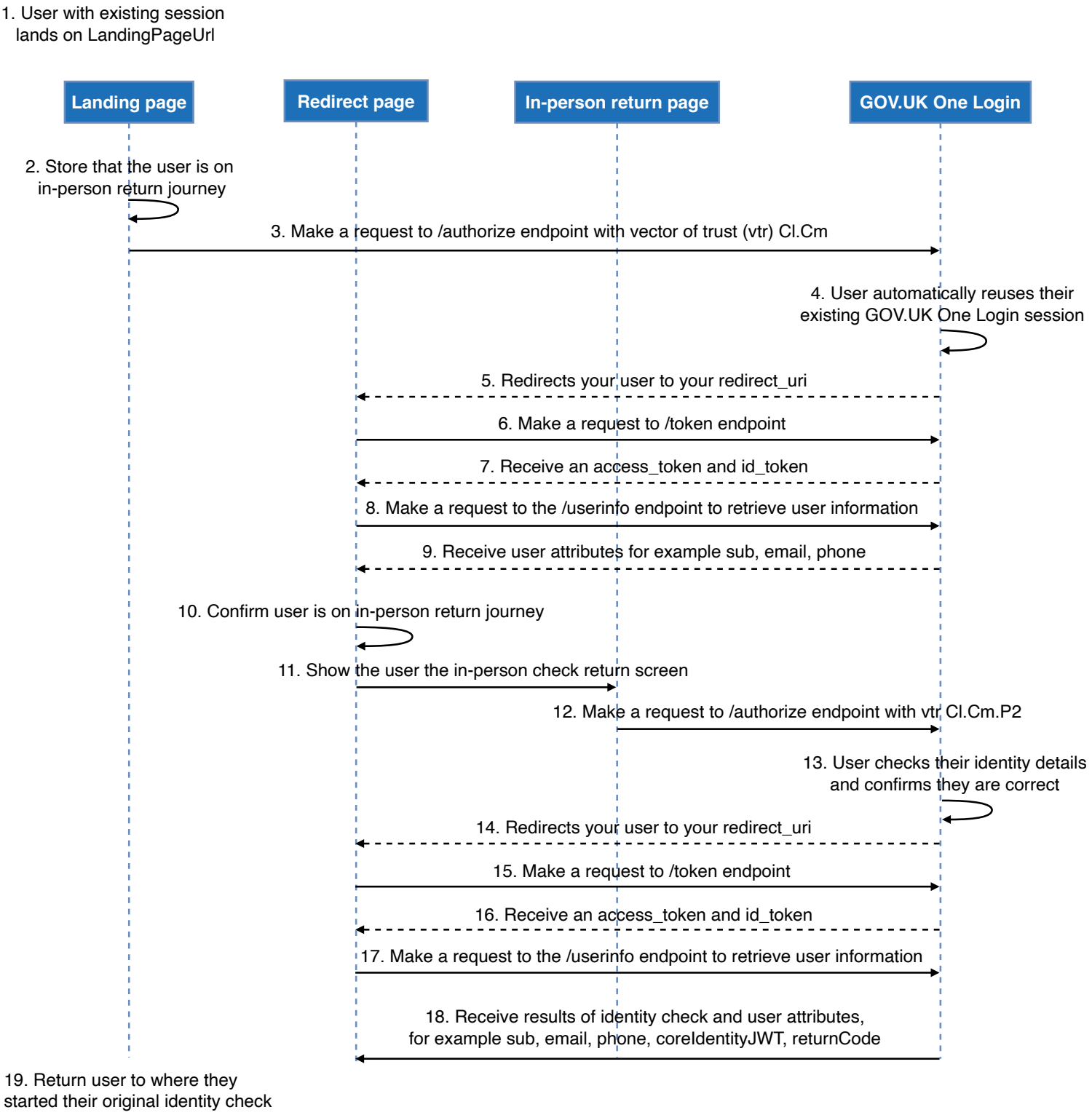
Understand what your landing page needs to do

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. 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) `Cl.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 `Cl.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.

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) 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
`id_token_hint`	Recommended - however, if you use `post_logout_redirect_uri` , this parameter is required	This is the ID token GOV.UK One Login previously issued when you made a request to the `/token` endpoint for your user's current session.
`post_logout_redirect_uri`	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 `id_token_hint` . This parameter is the URL you want to redirect your users to after you have logged them out. The `post_logout_redirect_uri`

		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 `post_logout_redirect_uri` .

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--lda%d
```

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).

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).

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_02YOlWpd8PAOI-2sVlB2nsNU7mcLZYhYw="
 "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).
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).
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).
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.

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_02YOlWpd8PAOI-2sVlB2nsNU7mcLZYhYw=",
  "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_02YOlWpd8PAOI-2sVlB2nsNU7mcLZYhYw=",
  "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": [
```

```
    {
      "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/). 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/).

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": "QrP65yghuglwPkEl11oMaabr4WqAMjuvztBYb7T4Ipo",
        "y": "CSQNybYbCZLl-Xr1OA3pcxjC6qZrG7JPqwXgo-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). 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:
    - 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).
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).

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 `Cl.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

`name`	A list showing the names proven by GOV.UK One Login. This list reflects name changes by using the `validFrom` and `validUntil` metadata properties. If `validUntil` is `null` or not present, that name is your user's current name. If `validFrom` is `null` or not present, your user may have used that name from birth. Each name is presented as an array in the `nameParts` property. Each part of the name is either a `GivenName` or a `FamilyName`, identified in its `type` property. The `value` property could be any text string. GOV.UK One Login cannot specify a maximum length or restrictions on what characters may appear. `GivenName` or `FamilyName` 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.
`birthDate`	A list of ISO 8601 date (https://schema.org/Date) 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

`validFrom`	ISO 8601 date (https://schema.org/Date) strings representing the date your user moved into the address. GOV.UK One Login only collects the year from the user. The month and day will always default to `01`. For example, if the user moved in 2024, the `validFrom` date would be `2024-01-01`.
`validUntil`	ISO 8601 date (https://schema.org/Date) strings representing the date your user moved from the address. This property is not included for your user's current address. GOV.UK One Login only collects the year from the user. The month and day will always default to `01`. For example, if the user moved in 2024, the `validUntil` date would be `2024-01-01`. If a user tells us an address is their current address, then `validUntil` will be not be returned.
`uprn`	GOV.UK One Login will provide a Unique Property Reference Number (UPRN) (https://www.gov.uk/government/publications/open-standards-for-government/identifying-property-and-street-information) for UK addresses only. If a user has edited their address, the UPRN field will automatically clear.
`organisationName`	Maps to `ORGANISATION_NAME` in the Postcode Address File (https://www.royalmail.com/find-a-postcode) and Ordnance Survey Places API (https://apidocs.os.uk/docs/os-places-dpa-output).
`departmentName`	Maps to `DEPARTMENT_NAME` in the Postcode Address File (https://www.royalmail.com/find-a-postcode) and Ordnance Survey Places API (https://apidocs.os.uk/docs/os-places-dpa-output).
`subBuildingName`	Maps to `SUB_BUILDING_NAME` in the Postcode Address File (https://www.royalmail.com/find-a-postcode) and Ordnance Survey Places API (https://apidocs.os.uk/docs/os-places-dpa-output). `subBuildingName` may accompany either `buildingName` or `buildingNumber`.
`buildingNumber`	Maps to `BUILDING_NUMBER` in the Postcode Address File (https://www.royalmail.com/find-a-postcode) and Ordnance Survey Places API (https://apidocs.os.uk/docs/os-places-dpa-output).

`buildingName`	Maps to `BUILDING_NAME` in the Postcode Address File (https://www.royalmail.com/find-a-postcode) and Ordnance Survey Places API (https://apidocs.os.uk/docs/os-places-dpa-output).
`dependentStreetName`	Maps to `DEPENDENT_THOROUGHFARE_NAME` in the Postcode Address File (https://www.royalmail.com/find-a-postcode) and Ordnance Survey Places API (https://apidocs.os.uk/docs/os-places-dpa-output).
`streetName`	Maps to `THOROUGHFARE_NAME` in the Postcode Address File (https://www.royalmail.com/find-a-postcode) and Ordnance Survey Places API (https://apidocs.os.uk/docs/os-places-dpa-output).
`doubleDependentAddressLocality`	Maps to `DOUBLE_DEPENDENT_LOCALITY` in the Postcode Address File (https://www.royalmail.com/find-a-postcode) and Ordnance Survey Places API (https://apidocs.os.uk/docs/os-places-dpa-output).
`dependentAddressLocality`	Maps to `DEPENDENT_LOCALITY` in the Postcode Address File (https://www.royalmail.com/find-a-postcode) and Ordnance Survey Places API (https://apidocs.os.uk/docs/os-places-dpa-output).
`addressLocality`	Maps to `POST_TOWN` in the Postcode Address File (https://www.royalmail.com/find-a-postcode) and Ordnance Survey Places API (https://apidocs.os.uk/docs/os-places-dpa-output).
`addressRegion`	Maps to schema:addressRegion (https://schema.org/addressRegion). 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 `POST_CODE` in the Postcode Address File (https://www.royalmail.com/find-a-postcode) and Ordnance Survey Places API (https://apidocs.os.uk/docs/os-places-dpa-output).
`addressCountry`	Two-letter ISO 3166-1 alpha-2 country code (https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2).

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.

Property	Definition

`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). 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 `returnCode` claim when you register your service.
2. Include `https://vocab.account.gov.uk/v1/returnCode` 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/returnCode` 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 `returnCode` 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 for more detailed information on what issue each `returnCode` value stands for.

Currently, there are 9 `returnCode` 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 for more detailed information on what each return code means.

Property	Definition
`code`	An array of single letter codes for `returnCode` 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 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 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_02YOlWpd8PAOI-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).

This page was last reviewed on 12 February 2025.

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.

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).
2. If you do not already have it, install Docker Desktop (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/).
4. Enable Docker Host networking (https://docs.docker.com/engine/network/drivers/host/#docker-desktop).
5. Install 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):
    - set **Prove user's identities** to **Yes**
    - set the claims to `coreIdentityJWT`, `returnCode` 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`, `returnCode` (empty), `address` claims. If you do not, get in touch

(https://docs.sign-in.service.gov.uk/support/).

If you have any issues:

- get in touch on the govuk-one-login-tech-support Slack channel (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

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

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`.
`/.well-known/openid-configuration`	An OpenID configuration endpoint.
`/.well-known/jwks.json`	A JSON Web Keys (JWKS) endpoint to publish the public keys that sign the ID token.
`/.well-known/did.json`	A decentralised identifier (DID) endpoint to publish the public keys that sign the core identity.
`/authorize`	An OpenID Connect (OIDC) endpoint to An OpenID Connect (OIDC) endpoint to authenticate the user (https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/authenticate-your-user/#make-a-request-to-the-authorize-endpoint).
`/config`	A simulator configuration endpoint for modifying and requesting the current configuration using `POST` and `GET`.
`/logout`	An OIDC endpoint to 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).
`/token`	An OIDC endpoint to exchange the authorisation code for tokens (https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/authenticate-your-user/#make-a-token-request).
`/trustmark`	A OIDC trustmark document listing vectors of trust implemented by GOV.UK One Login.
`/userinfo`	An OIDC endpoint to retrieve user information (https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/authenticate-your-user/#retrieve-user-information).

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/) (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).
2. If you do not already have it, install 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	HGIOgho9HIRhgoepdIOPFdIUWgewi0jw
publicKey	-----BEGIN PUBLIC KEY----- MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAmXXR3EsRvUMVhEJMtQ1wexJjfQ00Q0MQ7ARfShN53BnOQEPFnS/I8ntBddkKdE3q+vMTI72w6Fv3SsMM+ciR2LIHdEQfKgsLt6PGNcV1kG6GG/3nSW3psW8w65Q3fmy81P1748qezDrVfaGrF4PDXALzX1ph+nz8mpKmck6aY6LEUJ4B+TIfYzlKmmwFe3ri0spSW+J5wE9mmT3VkR2ySuHRYHQlxlF9dfX7ltOTsbgJFzN6TO01ZQDhY0iLwzdGwhSxO6R6N/ZINYHCKFPaQD+tdKsrw7QDIYnx0IiXFnkGnizl3UtqSmXAaceTvPM2Pz84x2JiwHrp2Sml6RYLCQIDAQAB -----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_02YOlWpd8PAOI-2sVlB2nsNU7mcLZYhYw=
email	test@example.com
emailVerified	true
phoneNumber	07123456789
phoneNumberVerified	true
maxLoCAchieved	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
simulatorUrl	http://localhost:3000

`postLogoutRedirectUrls`	`["http://localhost:8080/signed-out]`

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.	• `https://vocab.account.gov.uk/v1/passport` • `https://vocab.account.gov.uk/v1/address` • `https://vocab.account.gov.uk/v1/drivingPermit` • `https://vocab.account.gov.uk/v1/coreIdentityJWT` • `https://vocab.account.gov.uk/v1/returnCode`
`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_SUPPORTED`	`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 `client_assertion` signature.	PEM-encoded public key

`REDIRECT_URLS`	`redirectUrls`	The redirect URLs, which your users will be redirected to.	Any valid URLs
`SCOPES`	`scopes`	The scopes you've configured the client to request.	• `openid` • `email` • `phone`

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	`coreIdentityVerifiableCredentials`	A core identity verifiable credential.	JSON object
N/A	`drivingPermitDetails`	A set of driving licence details the simulator returns to the user.	JSON array
`EMAIL`	`email`	The returned email address.	Any string
`EMAIL_VERIFIED`	`emailVerified`	Whether or not the email address has been verified.	Boolean
N/A	`maxLoCAchieved`	The maximum level of confidence the user achieved.	Any string

N/A	`passportDetails`	A set of passport details the simulator returns to the user.	JSON array
`PHONE_NUMBER`	`phoneNumber`	The returned phone number.	Any string
`PHONE_NUMBER_VERIFIED`	`phoneNumberVerified`	Whether or not the phone number has been verified.	Boolean
N/A	`postalAddressDetails`	A set of address details the simulator returns to the user.	JSON array
N/A	`returnCodes`	A set of codes returned if the return code claim is included in the client configuration and `/authorize` request. Otherwise an `ACCESS_DENIED` error will return when this is configured.	JSON array with the following structure `[{"code": "anyString"}]`
`SUB`	`sub`	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).

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 Authenticate your user (https://docs.sign-in.service.gov.uk/integrate-with-integration-environment/authenticate-your-user/#http-302-found) 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 (`vot`) returned in the token does not match the vector of trust requested (`vtr`) in the `/authorize` request.
`INVALID_ALG_HEADER`	The `alg` in the header does not match the algorithm returned from the `/jwks` 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_EXPIRED`	The expiry date of the token is in the past.
`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
`INCORRECT_SUB`	The `sub` does not match the `sub` in the `id_token`. Sub is the 'subject identifier' or the unique ID of a user.
`INVALID_ALG_HEADER`	The `alg` in the header is not `ES256`.
`INVALID_AUD`	Core identity has an invalid audience.
`INVALID_ISS`	Core identity has an invalid issuer.
`INVALID_SIGNATURE`	The signature of the token is invalid.
`TOKEN_EXPIRED`	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).

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) or in the 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) 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).

This page was last reviewed on 12 June 2025.

View source Report problem GitHub Repo

Accessibility

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` . 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` . 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` . 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` . 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), 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` . 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

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. Email GOV.UK One Login to access this test user data.

Supports `client_secret_post`	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 configuring the GOV.UK One Login simulator (/test-your-integration/gov-uk-one-login-simulator/#configure-the-gov-uk-one-login-simulator) for more information.	Yes
Supports performance testing	Yes - in `interactive_mode` (/test-your-integration/gov-uk-one-login-simulator/#returning-multiple-response-configurations) 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

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 "Setting a User-Agent header on HTTP requests" (/before-integrating/set-user-agent-header) the requirement to use an appropriate `User-Agent` header on service calls to GOV.UK One Login.
Sep 2 2025	Updates guidance "Prove your user's identity" (/integrate-with-integration-environment/prove-users-identity/#prove-your-user-39-s-identity) with guidance for helping your users after their in-person identity checks.

Jul 30 2025	Updates guidance 'Choose which scopes your service can request' (/before-integrating/choose-which-user-attributes-your-service-can-request/#choose-which-scopes-your-service-can-request) and 'Retrieve user information' (/integrate-with-integration-environment/authenticate-your-user/#receive-response-for-retrieve-user-information) to add information about the `wallet-subject-id` scope.
Jun 12 2025	Updates section on testing to remove guidance on building mocks and move guidance on "using the GOV.UK One Login simulator" (/test-your-integration/gov-uk-one-login-simulator/) to section on "testing your integration with GOV.UK One Login" (/test-your-integration/).
May 2 2024	Updates guidance to add information about using Proof Key for Code Exchange (PKCE) parameters in the authorise request. (/integrate-with-integration-environment/authenticate-your-user/#make-a-request-to-the-authorize-endpoint) Updates guidance to include information about using PKCE parameters in the token request. (/integrate-with-integration-environment/authenticate-your-user/#make-a-token-request) Updates guidance to include guidance about `PKCEEnforced` field when configuring your service (/configure-for-production/).
Apr 15 2025	Updates guidance "Managing user sessions if your service session is less than 1 hour" (/integrate-with-integration-environment/managing-your-users-sessions/#managing-user-sessions-if-your-service-session-is-less-than-1-hour) to add guidance on how to re-authenticate your users. Updates the 'Make a request to the /authorize endpoint' table (/integrate-with-integration-environment/authenticate-your-user/#error-handling-for-make-a-request-to-the-authorize-endpoint) to add an entry for `login_required` error code.
Apr 2 2025	New guidance "Test your service with the GOV.UK One Login simulator" (/test-your-integration/gov-uk-one-login-simulator/) to add information about the new GOV.UK One Login simulator.
Mar 5 2025	Updates guidance "Integrating third-party platforms with GOV.UK One Login" (/before-integrating/integrating-third-party-platform/#integrating-third-party-platforms-with-gov-uk-one-login) to add guidance on integrating with GOV.UK One Login using Amazon Cognito.
Feb 17 2025	Updates guidance "Using the integration environment for end-to-end testing" (/test-your-integration/using-integration-for-testing/#using-the-integration-environment-for-end-to-end-testing) to remove reference to the integration environment basic authentication challenge which has been removed and is no longer required.
Jan 27 2025	Updates guidance "Authenticate your user" (/integrate-with-integration-environment/authenticate-your-user) 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 "Quick start" (/quick-start/) to help users see how a typical integration with GOV.UK One Login works.
Oct 23 2024	Updates guidance "Understand the core identity signing key rotations" (/integrate-with-integration-environment/prove-users-identity/#understand-the-core-identity-signing-key-rotations) 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 "share your public keys using a JWKS endpoint" to add other option when sharing your public key with GOV.UK One Login.
Sep 25 2024	Updates guidance "Register and manage your service" (/before-integrating/register-and-manage-your-service/#register-and-manage-your-service) to add guidance on how to register and manage a service.
Sep 17 2024	Updates guidance "Integrating third-party platforms with GOV.UK One Login" (/before-integrating/integrating-third-party-platform/#integrating-third-party-platforms-with-gov-uk-one-login) to add guidance on integrating with GOV.UK One Login using Salesforce.
Sep 6 2024	Updates guidance "Use the production discovery endpoint" (/configure-for-production/#use-the-production-discovery-endpoint) to add the production discovery endpoint.
Aug 21 2024	Updates guidance "Configure your service for production" (/configure-for-production/) to add information about how to configure your service for production.
Aug 20 2024	Updates guidance "Receive response for 'Retrieve user information'" (/integrate-with-integration-environment/authenticate-your-user/#receive-response-for-retrieve-user-information) to add a table explaining more about the response from the `/userinfo` endpoint.
Jul 29 2024	Updates guidance "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) to update we now return HTTP 400 Bad Request errors for requests with incorrect parameters.
Jul 18 2024	New guidance "Validate 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). 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 `https://vocab.account.gov.uk/v1/socialSecurityRecord` claim
Jul 4 2024	New guidance "Integrating third-party platforms with GOV.UK One Login" (/before-integrating/integrating-third-party-platform/) which contains information about integrating with GOV.UK One Login using a third-party platform, and contains details about the `client_secret_post` token authentication method.
Jun 21 2024	Updates guidance "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) to clarify the `{"message": "Internal server error"}` HTTP 502 Bad gateway error.
Jun 18 2024	Includes example data to help with building mocks: Access example data.
May 22 2024	New guidance Using the integration environment for end-to-end testing (/test-your-integration/using-integration-for-testing/) 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 Managing your users' sessions (/integrate-with-integration-environment/managing-your-users-sessions/) to explain how to manage your users' sessions and how to build a logout mechanism for your users.
Apr 9 2024	Updates the technical flow diagram (/how-gov-uk-one-login-works/#understand-the-technical-flow-gov-uk-one-login-uses) to document the use of the `/logout` endpoint.
Apr 3 2024	New guidance Understand your user's return code claim (/integrate-with-integration-environment/prove-users-identity/#understand-your-user-s-return-code-claim) 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 `offline_access` to simplify integration and the technical flow.
Feb 14 2024	New guidance Choose your sector identifier (/before-integrating/choose-your-sector-identifier/) to explain the use of the sector identifier with a

	worked example that shows the effect of choosing different sector identifiers.
Dec 22 2023	Updates guidance on making a request to the `/authorize` endpoint.
Dec 21 2023	New guidance Secure your authorisation request parameters with JWT (/integrate-with-integration-environment/authenticate-your-user/#secure-your-authorisation-request-parameters-with-jwt) 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 Before you integrate with GOV.UK One Login (/before-integrating).

This page was last reviewed on 17 September 2024.

View source Report problem GitHub Repo

Accessibility