OAuth2 / Keycloak

with ASP.NET Core (C#) and Angular

Donaueschingen, October 2023

Intro

Who am I?

Florian Schick

// Independent Software Developer //

SCHICK

SOFTWARE ENTWICKLUNG

Focus on

Full-Stack with .NET/Core, C#, Angular, Vue.js

Clean code // easy to read, easy to maintain //

Contact

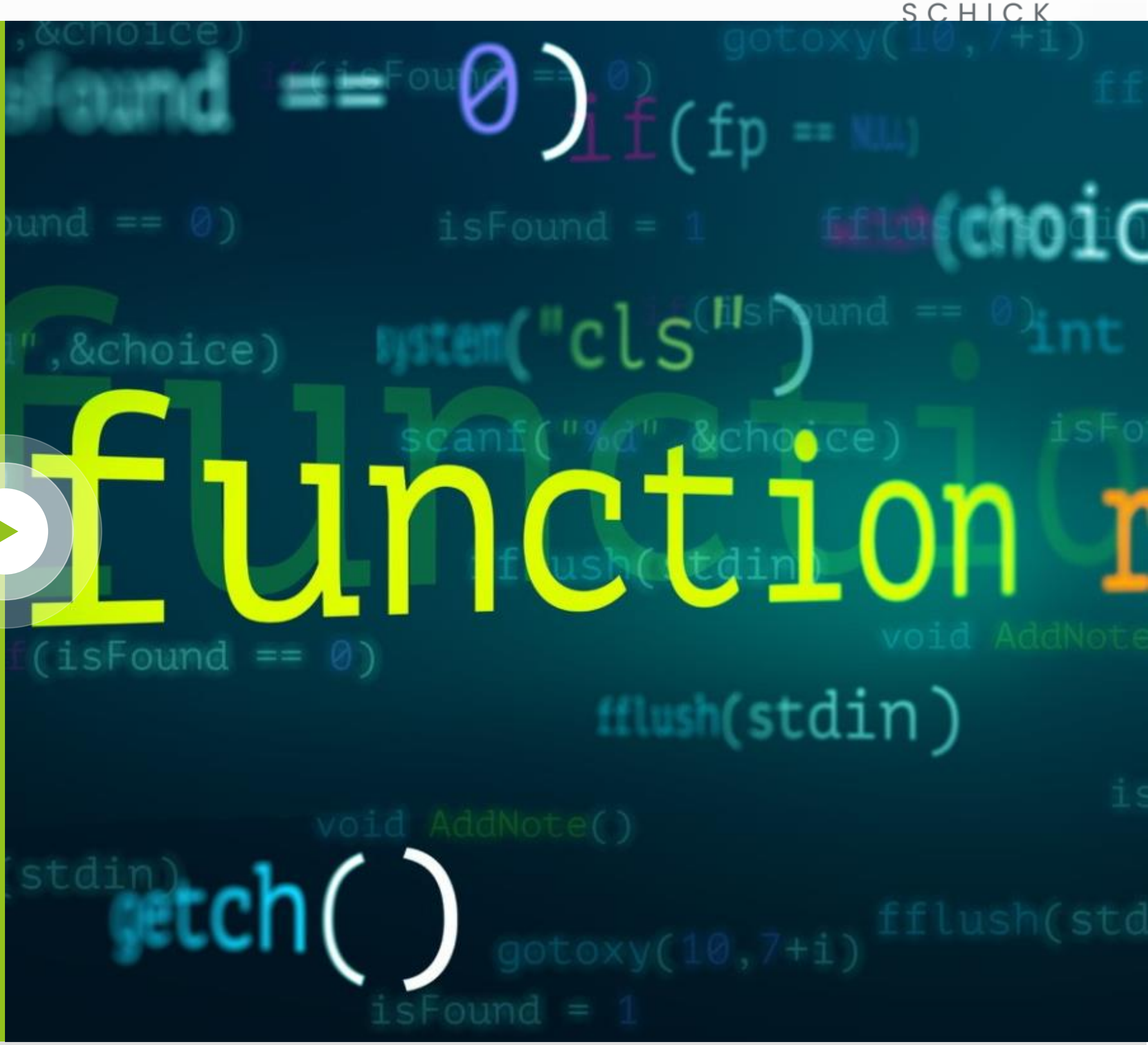
☎ florian.schick@schick-software.de

@ +49 771 8979378

Intro

Agenda

1. Explanation of terms
2. Install and configure Keycloak
3. Authentication flows
4. Authentication
5. Identity providers
6. Authorization (role based)
7. Authorization (UMA based)



Explanation of terms

OAuth2 / OpenID Connect

OAuth2 (Open Authorization)

Is an open standard to give websites or applications access to user's information without having to give them their passwords

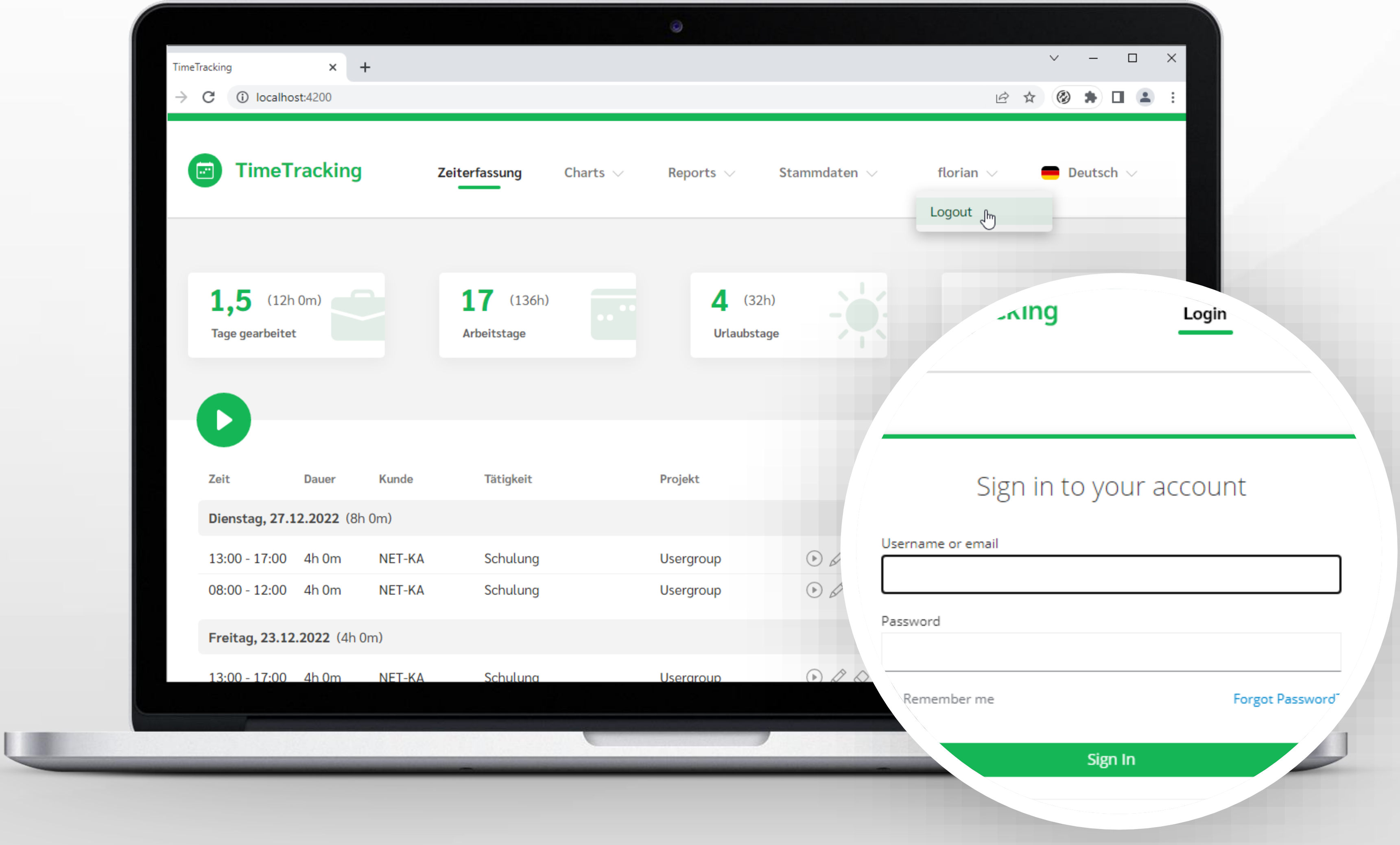
OpenID Connect (OIDC)

Is an authentication layer on top of the OAuth 2.0 authorization framework. It allows websites or applications to verify the identity of an end user based on the authentication performed by an authorization server

Explanation of terms

Keycloak

Keycloak is an open source software product to allow **single sign-on with Identity and Access Management** aimed at modern applications and services.



1,5 (12h 0m) Tage gearbeitet

17 (136h) Arbeitstage

4 (32h) Urlaubstage

Zeit	Dauer	Kunde	Tätigkeit	Projekt
Dienstag, 27.12.2022 (8h 0m)				
13:00 - 17:00	4h 0m	NET-KA	Schulung	Usergroup
08:00 - 12:00	4h 0m	NET-KA	Schulung	Usergroup
Freitag, 23.12.2022 (4h 0m)				
13:00 - 17:00	4h 0m	NET-KA	Schulung	Usergroup

Sign in to your account

Username or email

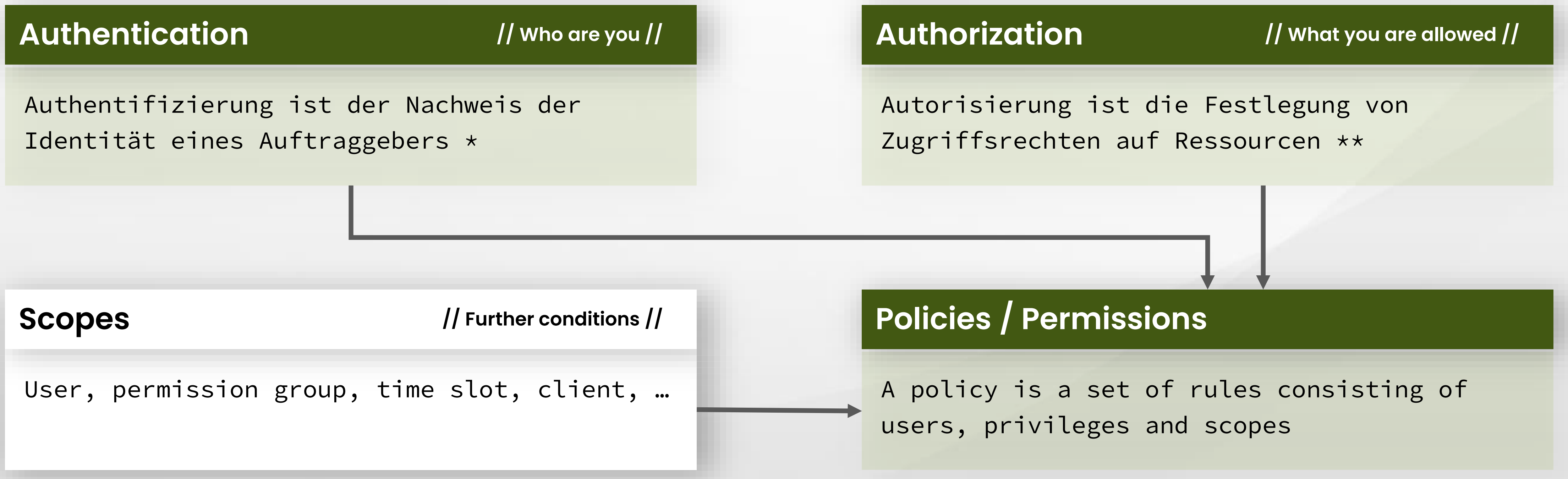
Password

Remember me

Forgot Password

Explanation of terms

Authentication ≠ Authorization



Authentication // Who are you //

Authentifizierung ist der Nachweis der Identität eines Auftraggebers *

Authorization // What you are allowed //

Autorisierung ist die Festlegung von Zugriffsrechten auf Ressourcen **

Scopes // Further conditions //

User, permission group, time slot, client, …

Policies / Permissions

A policy is a set of rules consisting of users, privileges and scopes

* Principal is the identity of a user, computer system, device, …

** Resource is a piece of information like customers, documents, …

Explanation of terms

Names like smoke and mirrors

Resource Identifier

Resource

Client

ClientId

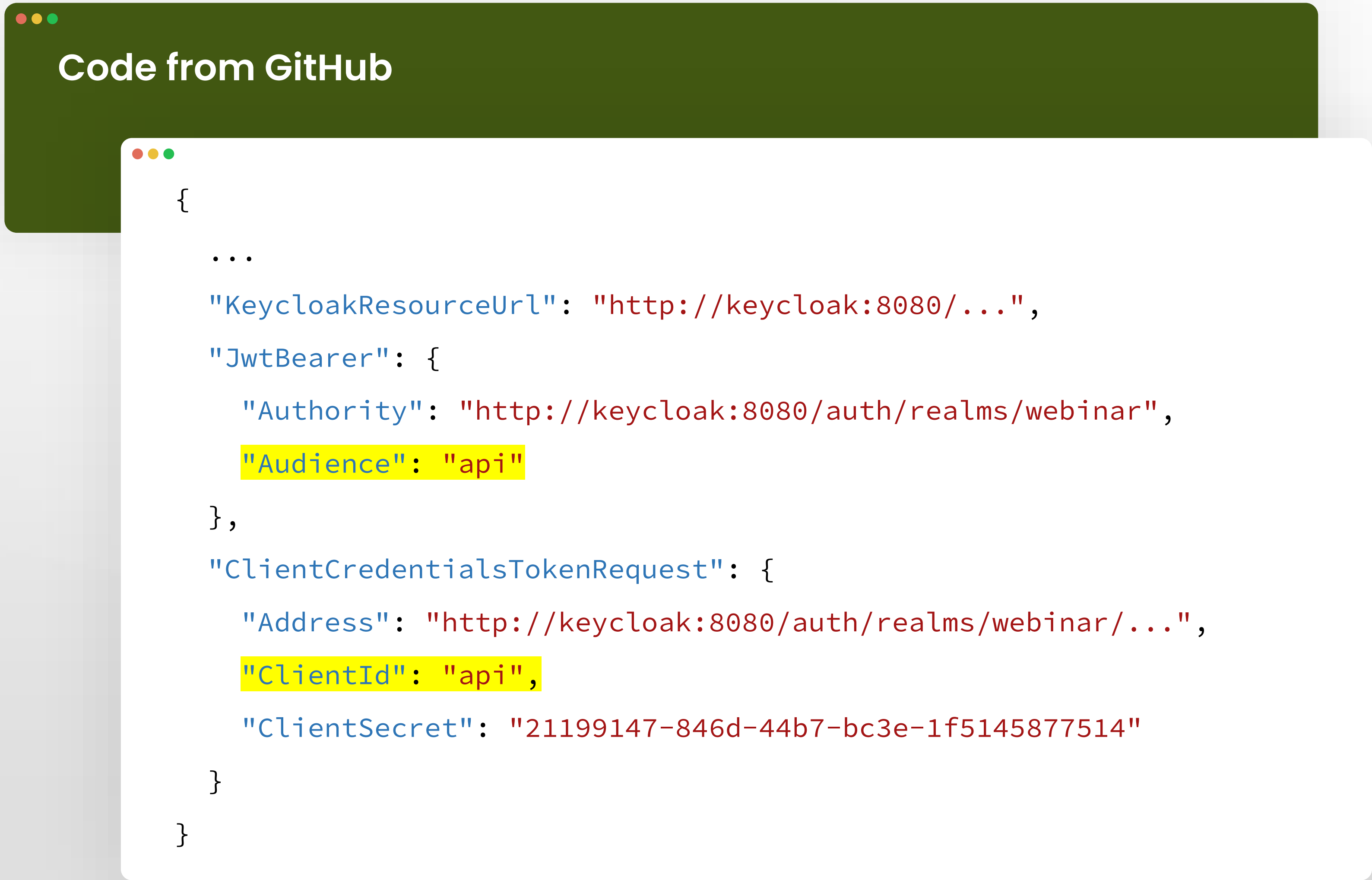
Resource Server

Audience

Identify what Keycloak means by a "client"

Explanation of terms

Live sample



Code from GitHub

```
{
  ...
  "KeycloakResourceUrl": "http://keycloak:8080/...",
  "JwtBearer": {
    "Authority": "http://keycloak:8080/auth/realms/webinar",
    "Audience": "api"
  },
  "ClientCredentialsTokenRequest": {
    "Address": "http://keycloak:8080/auth/realms/webinar/...",
    "ClientId": "api",
    "ClientSecret": "21199147-846d-44b7-bc3e-1f5145877514"
  }
}
```

OAuth2 / Keycloak

Install Keycloak

Authentication flows

Authentication

Identity providers

Authorization (role based)

Authorization (UMA based)

Install Keycloak

Authentication flows

Authentication

Identity providers

Authorization (role based)

Authorization (UMA based)

Install Keycloak

Install

- Install OpenJDK
- Download Keycloak Zip-Deployment
- Run bin/kc.sh (start-dev|start)

Modes

```
start-dev: HTTP, HTTPS
start : nur HTTPS
```

OAuth2 / Keycloak

Install Keycloak

Authentication flows

Authentication

Identity providers

Authorization (role based)

Authorization (UMA based)

Configure Keycloak

Realm

- A realm manages a set of users, credentials, roles, and groups
- A user belongs to and logs into one realm
- Realms are isolated from one another

Client

- Clients are applications and services that can request authentication of a user
- Clients can act as a resource server

User

Users are the users in the current realm

OAuth2 / Keycloak

Install Keycloak

Authentication flows

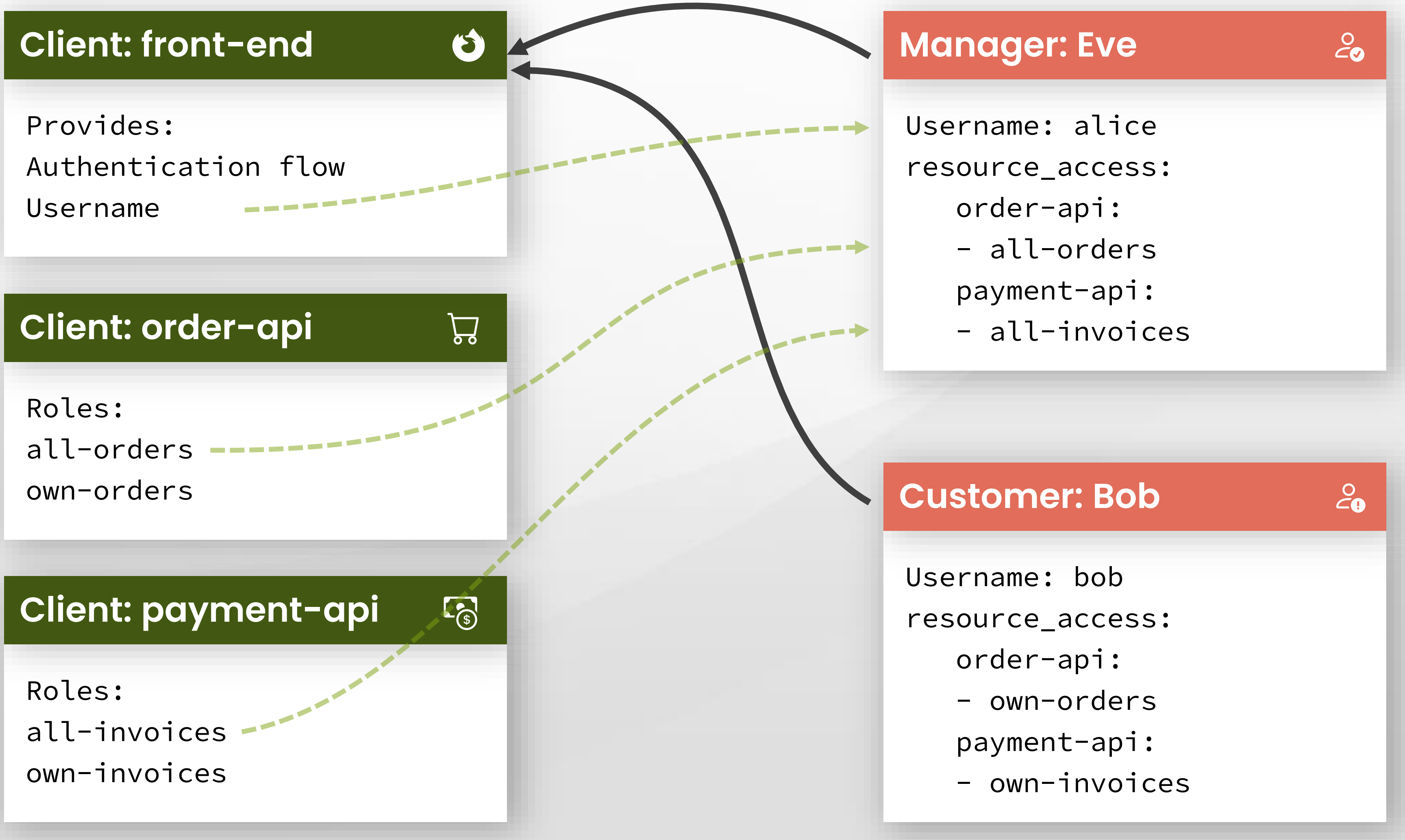
Authentication

Identity providers

Authorization (role based)

Authorization (UMA based)

Relation between clients and users



Client: front-end

```
Provides:
Authentication flow
Username
```

Client: order-api

```
Roles:
all-orders
own-orders
```

Client: payment-api

```
Roles:
all-invoices
own-invoices
```

Manager: Eve

```
Username: alice
resource_access:
   order-api:
   - all-orders
   payment-api:
   - all-invoices
```

Customer: Bob

```
Username: bob
resource_access:
   order-api:
   - own-orders
   payment-api:
   - own-invoices
```

OAuth2 / Keycloak

Install Keycloak

Authentication flows

Authentication

Identity providers

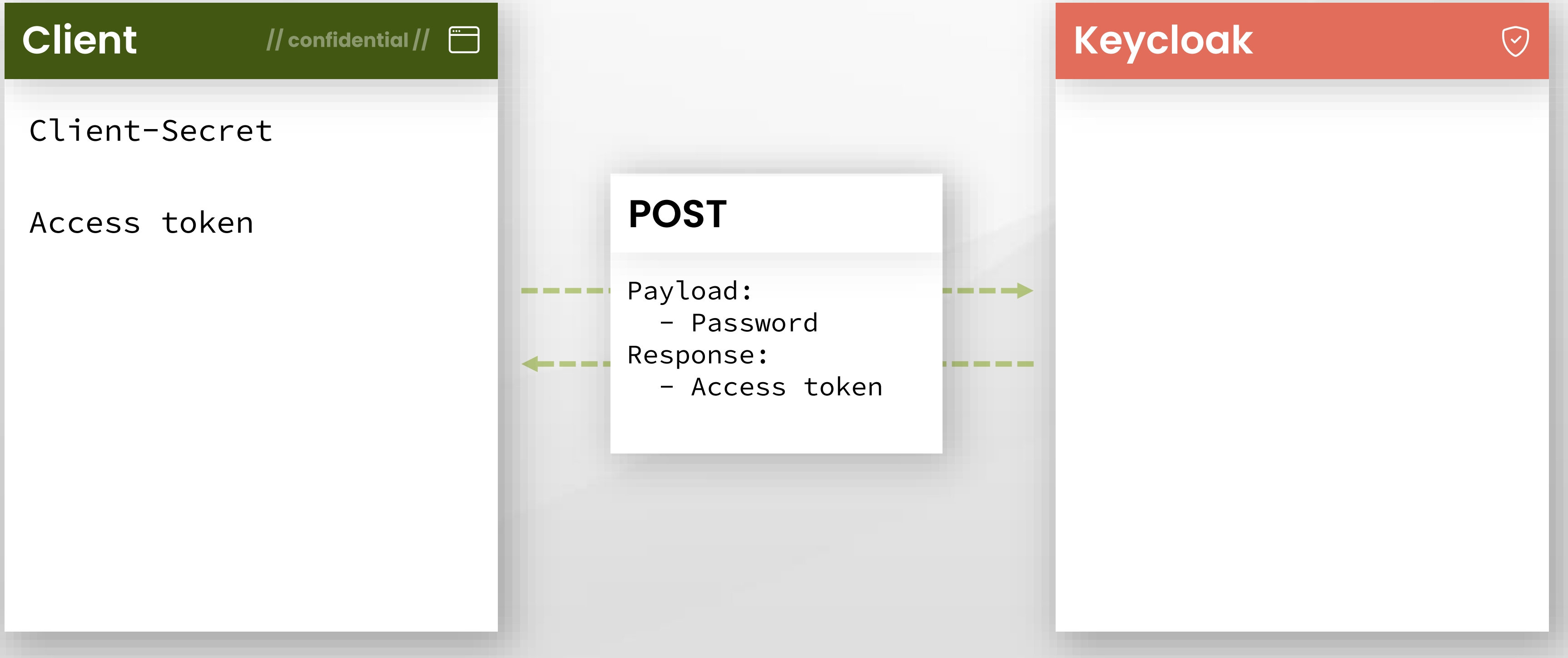
Authorization (role based)

Authorization (UMA based)

Authentication flows

// simplified //

Client Credentials Grant / Service accounts roles



Client // confidential //

```
Client-Secret

Access token
```

POST

```
Payload:
  - Password
Response:
  - Access token
```

Keycloak

OAuth2 / Keycloak

Install Keycloak

Authentication flows

Authentication

Identity providers

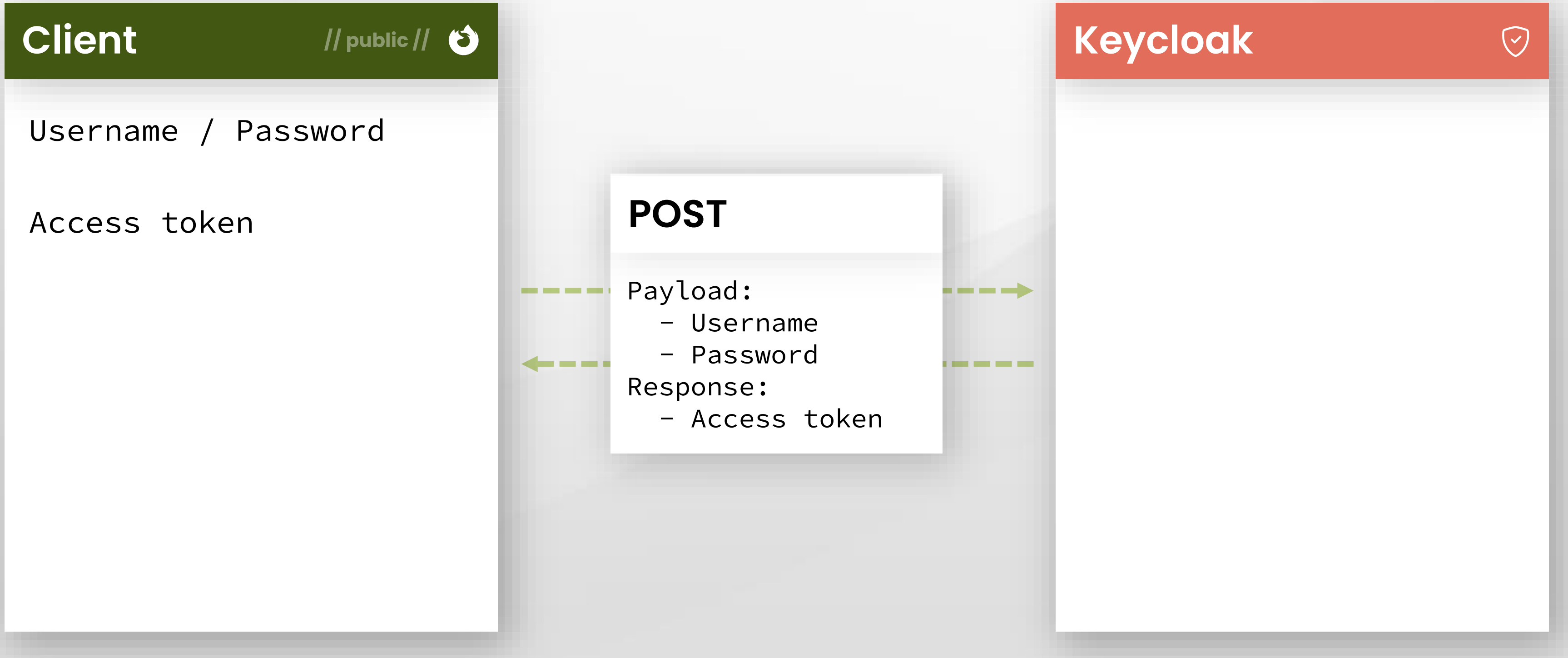
Authorization (role based)

Authorization (UMA based)

Authentication flows

// simplified //

Resource Owner Password Credentials Grant / Direct Access Grant



Username / Password

Access token

```
Payload:
  - Username
  - Password
Response:
  - Access token
```

OAuth2 / Keycloak

Install Keycloak

Authentication flows

Authentication

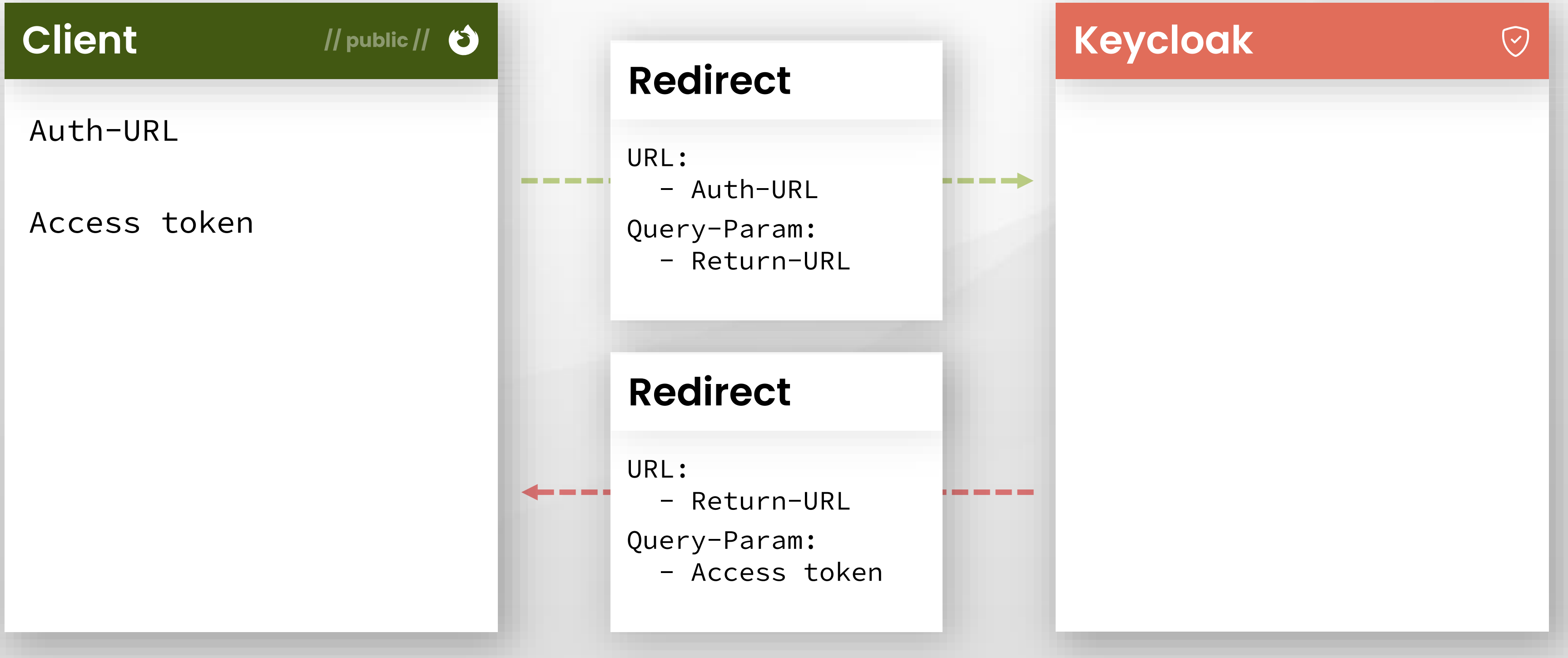
Identity providers

Authorization (role based)

Authorization (UMA based)

Authentication flows
// simplified //

Implicit Flow // veraltet //



Client // public //

Auth-URL

Access token

Redirect

URL:
 - Auth-URL
Query-Param:
 - Return-URL

Redirect

URL:
 - Return-URL
Query-Param:
 - Access token

Keycloak

OAuth2 / Keycloak

Install Keycloak

Authentication flows

Authentication

Identity providers

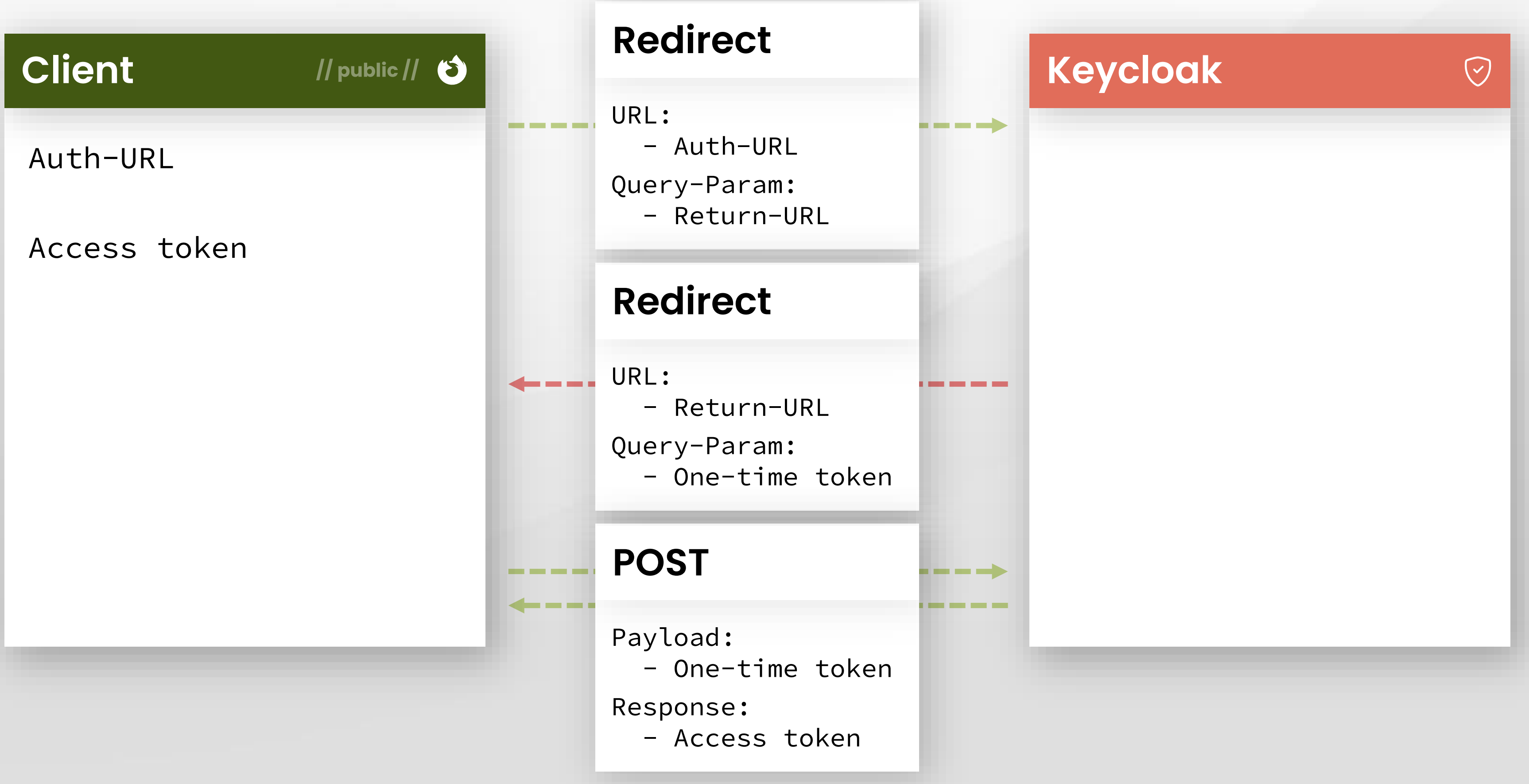
Authorization (role based)

Authorization (UMA based)

Authentication flows

// simplified //

Authorization Code Flow / Standard flow



Client // public //

Auth-URL

Access token

Redirect

URL:
  - Auth-URL
Query-Param:
  - Return-URL

Redirect

URL:
  - Return-URL
Query-Param:
  - One-time token

POST

Payload:
  - One-time token
Response:
  - Access token

Keycloak

OAuth2 / Keycloak

Install Keycloak

Authentication flows

Authentication

Identity providers

Authorization (role based)

Authorization (UMA based)

Authentication / ASP.NET

Enable authentication for controller routes

```
[Authorize], [AllowAnonymous]
```

Enable authentication for minimal API routes

```
webApplication.MapGet(...).RequireAuthorization(...)
webApplication.MapGet(...).AllowAnonymous()
```

OAuth2 / Keycloak

Install Keycloak

Authentication flows

Authentication

Identity providers

Authorization (role based)

Authorization (UMA based)

Authentication / ASP.NET

Register services

```
serviceCollection.AddAuthentication()
serviceCollection.AddJwtBearer(options => ...)
```

Configure application

```
webApplication.UseAuthentication();
webApplication.UseAuthorization();
```

OAuth2 / Keycloak

Install Keycloak

Authentication flows

Authentication

Identity providers

Authorization (role based)

Authorization (UMA based)

Authentication / OpenAPI

SCHICK
SOFTWARE ENTWICKLUNG

Register services

```
swaggerGenOptions.AddGenericAuthorization();
swaggerGenOptions.AddAuthorizationCodeFlow();

swaggerUIOptions.OAuthClientId("api");
swaggerUIOptions.OAuthUsePkce();
swaggerUIOptions.EnablePersistAuthorization();
```

OAuth2 / Keycloak

Install Keycloak

Authentication flows

Authentication

Identity providers

Authorization (role based)

Authorization (UMA based)

Authentication / Angular

SCHICK
SOFTWARE ENTWICKLUNG

Initialize Keycloak on startup

```
provide: APP_INITIALIZER,
useFactory: () => () => new Keycloak(<options>).init(...)
```

Register HTTP interceptor

```
provide: HTTP_INTERCEPTORS,
useClass: AuthenticationInterceptor,
```

OAuth2 / Keycloak

Install Keycloak

Authentication flows

Authentication

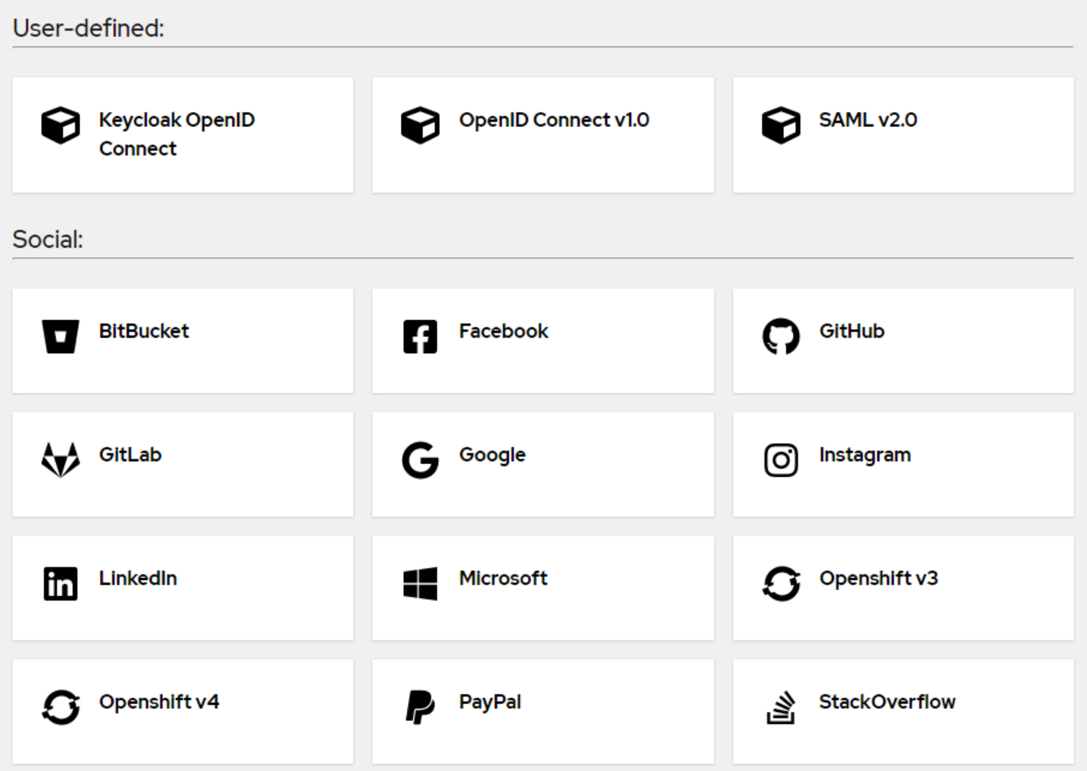
Identity providers

Authorization (role based)

Authorization (UMA based)

Identity providers

Demo: Add an identity provider

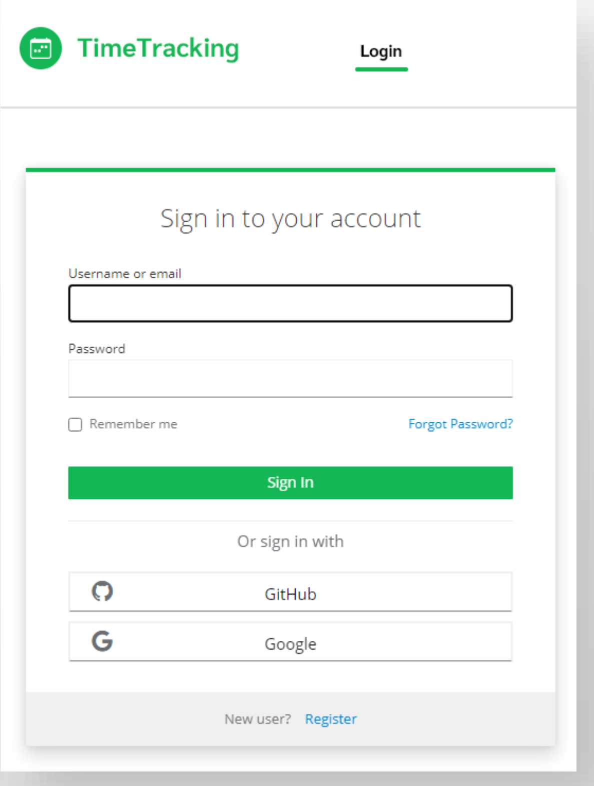


User-defined:

- Keycloak OpenID Connect
- OpenID Connect v1.0
- SAML v2.0

Social:

- BitBucket
- Facebook
- GitHub
- GitLab
- Google
- Instagram
- LinkedIn
- Microsoft
- Openshift v3
- Openshift v4
- PayPal
- StackOverflow



TimeTracking

Login

Sign in to your account

Username or email

Password

□ Remember me

Forgot Password?

Sign In

Or sign in with

GitHub

Google

New user? Register

OAuth2 / Keycloak

Install Keycloak

Authentication flows

Authentication

Identity providers

Authorization (role based)

Authorization (UMA based)

Authorization / ASP.NET

Enable authorization for controller routes

```
[Authorize(Roles = "Manager")]
```

Enable authorization for minimal API routes

```
webApplication.MapGet(...).RequireAuthorization(...)
```

Register services

```
serviceCollection.AddAuthorization()
```

OAuth2 / Keycloak

Install Keycloak

Authentication flows

Authentication

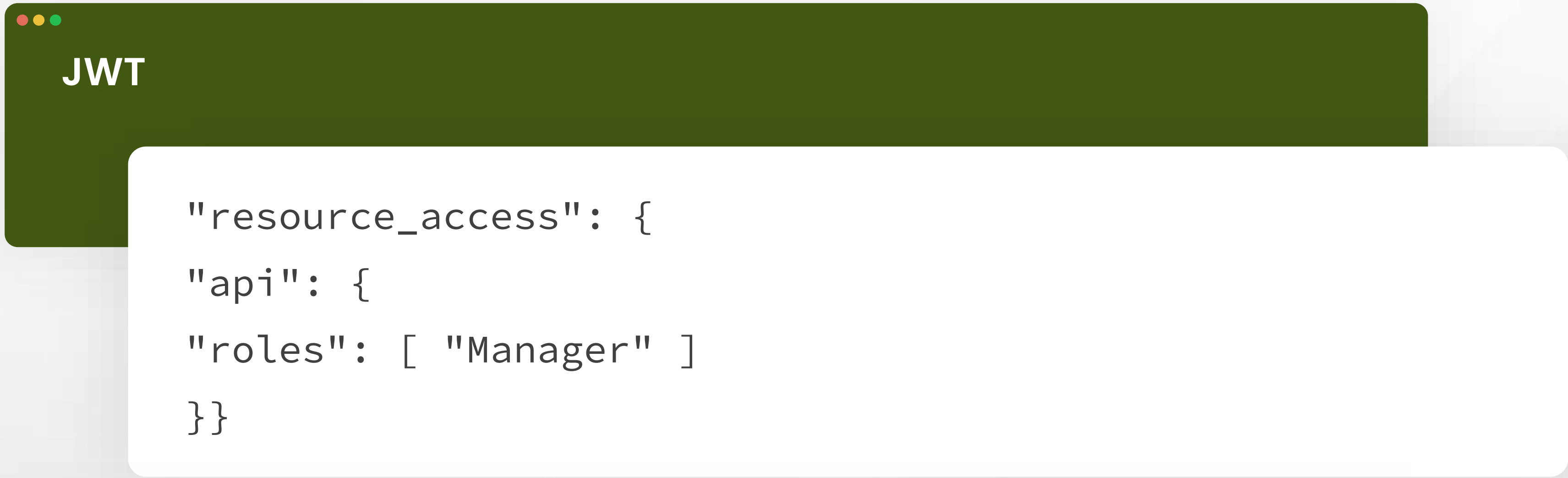
Identity providers

Authorization (role based)

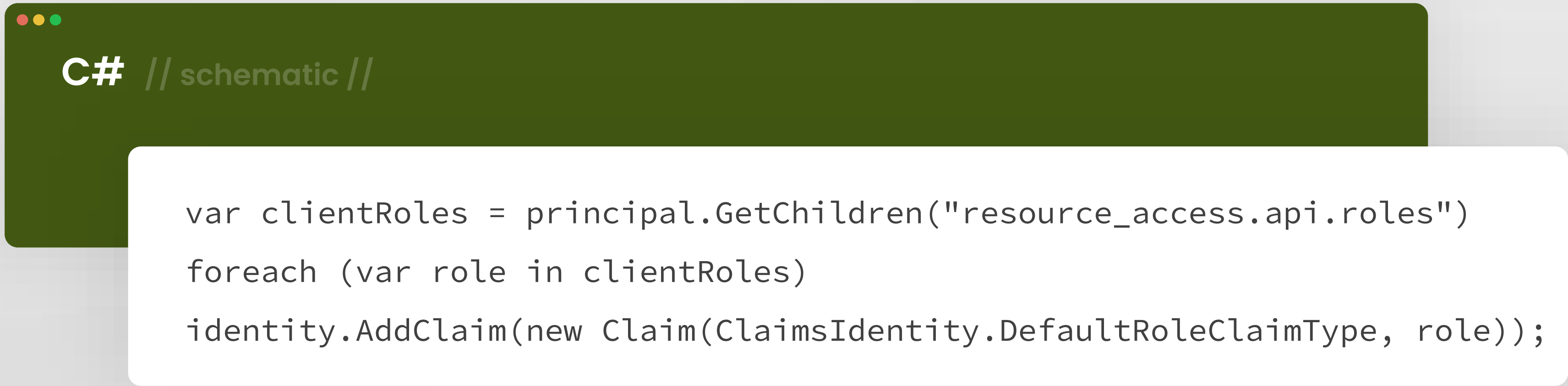
Authorization (UMA based)

Keycloak
Role transformation

SCHICK
SOFTWARE ENTWICKLUNG


JWT

```
"resource_access": {
"api": {
"roles": [ "Manager" ]
}}
```


C# // schematic //

```
var clientRoles = principal.GetChildren("resource_access.api.roles")
foreach (var role in clientRoles)
identity.AddClaim(new Claim(ClaimsIdentity.DefaultRoleClaimType, role));
```

OAuth2 / Keycloak

Install Keycloak

Authentication flows

Authentication

Identity providers

Authorization (role based)

Authorization (UMA based)

UMA
// User Managed Access protocol //

UMA

Coming as an OAuth-based access management protocol standard, UMA moves the process of creating and handling authorization rules from application to an identity server.

Install Keycloak

Authentication flows

Authentication

Identity providers

Authorization (role based)

Authorization (UMA based)

UMA
Keycloak

Enable UMA

Activate “Client authentication” and “Authorization” on the client acts as resource server

Anpassen der Standard-Policy

The default policy relies on JavaScript which isn’t shipped with Java anymore (Java > 15.x)

- Either get ‘Nashorn’ JS engine working with your Java installation
- Replace the default policy by something not using JS
- Use the docker image of Keycloak

OAuth2 / Keycloak

Install Keycloak

Authentication flows

Authentication

Identity providers

Authorization (role based)

Authorization (UMA based)

UMA Keycloak

Create authorization resources

Resources: Data to protected like customers, articles, …

Scopes: Actions to perform like read, create, update, delete, …

Policies: Conditions like ‘within business hours’, ‘is in Group’, …

Permissions: Which policy is allowed to perform which action for a resource?

Create client for front-end channel

The client ‘api’ became a confidential client requires a client secret. To keep the authorization working with angular, we need a new client ‘frontend’.

Sources

Ressourcen

Demo Applikation

https://github.com/fschick/Keycloak.ASPNet.Angular

Real-World Applikation with Keycloak

https://github.com/fschick/TimeTracking

Keycloak REST API client

https://github.com/fschick/Keycloak.RestApiClient

ASP.NET One-Time Token Authentication

https://github.com/fschick/Authentication.OneTimeToken

Sources

Ressourcen

An Illustrated Guide to OAuth and OpenID Connect

https://developer.okta.com/blog/2019/10/21/illustrated-guide-to-oauth-and-oidc

Background information to OAuth and OpenID Connect

LDAP Wiki OAuth 2.0
LDAP Wiki OpenID Connect

Use Keycloak as Identity Provider in ASP.NET Core 6

https://nikiforovall.github.io/aspnetcore/dotnet/2022/08/24/dotnet-keycloak-auth.html

Thank you

www.schick-software.de