OBJECT COMPUTING
HOME TO MICRONAUT®

SPAs with Svelte and Micronaut

Mark Volkmann and Sergio del Amo

MICRONAUT®

Object Computing, Inc. (OCI)
12140 Woodcrest Exec. Dr., Ste. 300
Saint Louis, MO 63141 USA

Limits of Liability and Disclaimer of Warranty

While every precaution has been taken in preparing this material, including research, development and testing, OCI assumes no responsibility for errors or omissions. No liability is assumed by OCI for any damages resulting from the use of this information.

Agenda

- Introductions
- OAuth 2.0 OpenID Connect - Auth
- API Documentation
- API Versioning
- Svelte
- Q&A
- Resources

OBJECT COMPUTING
HOME TO MICRONAUT®

OAuth 2.0 OpenID Connect Authentication

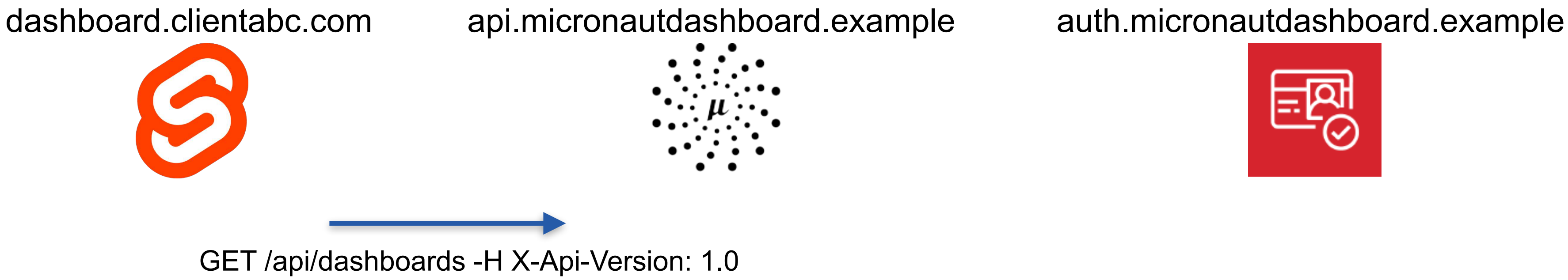
Amazon Cognito

Amazon Cognito User Pools is a standards-based Identity Provider and supports identity and access management standards, such as Oauth 2.0, SAML 2.0, and OpenID Connect.

dashboard.clientabc.com

api.micronautdashboard.example

auth.micronautdashboard.example

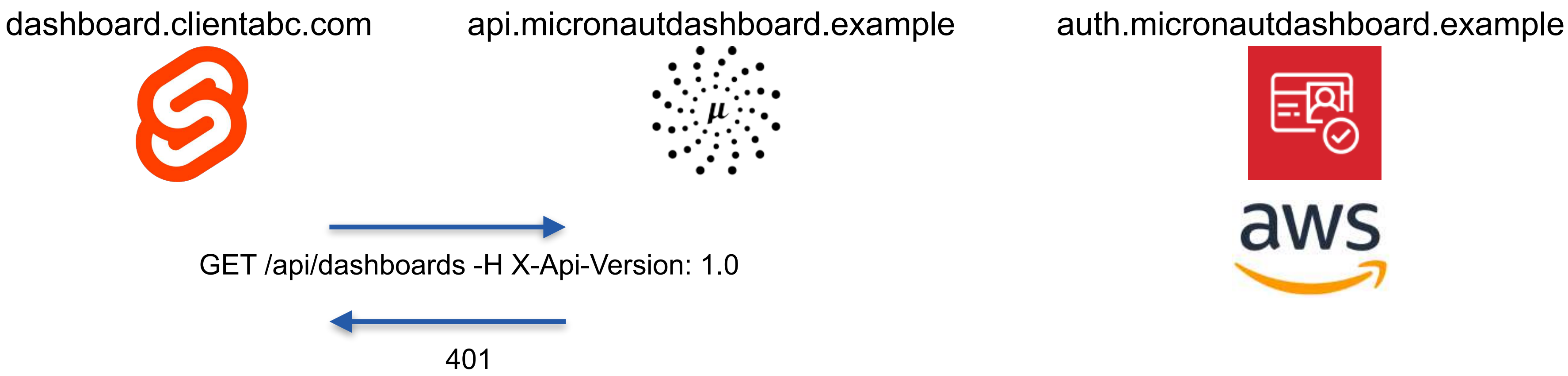


dashboard.clientabc.com

api.micronautdashboard.example

auth.micronautdashboard.example

GET /api/dashboards -H X-Api-Version: 1.0



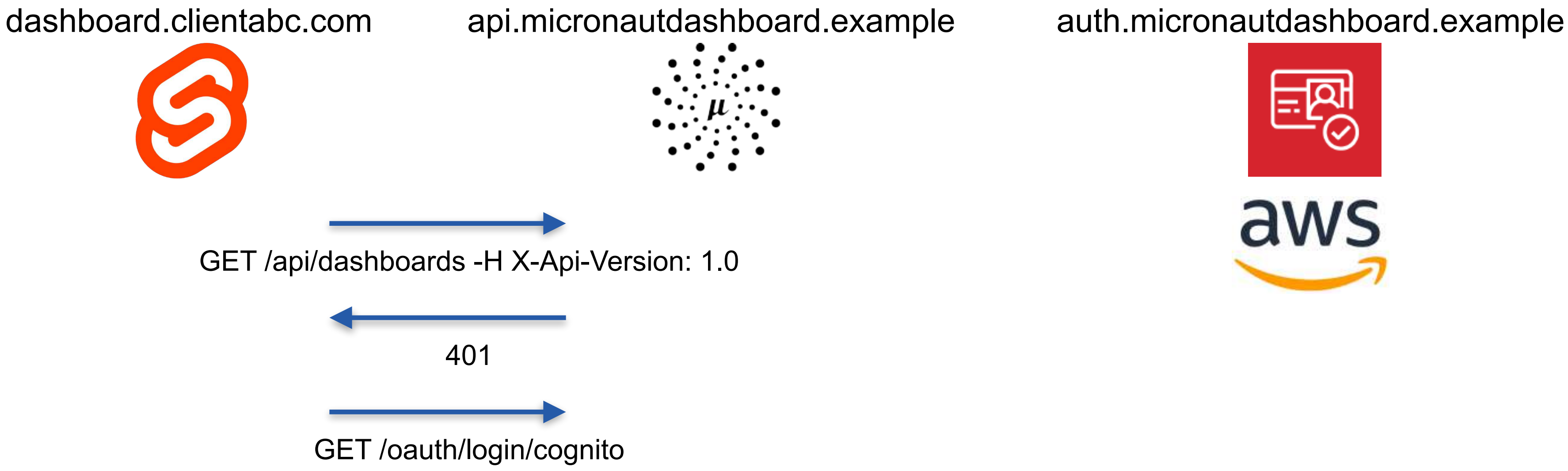
dashboard.clientabc.com

api.micronautdashboard.example

auth.micronautdashboard.example

GET /api/dashboards -H X-Api-Version: 1.0

401



dashboard.clientabc.com

api.micronautdashboard.example

auth.micronautdashboard.example

GET /api/dashboards -H X-Api-Version: 1.0

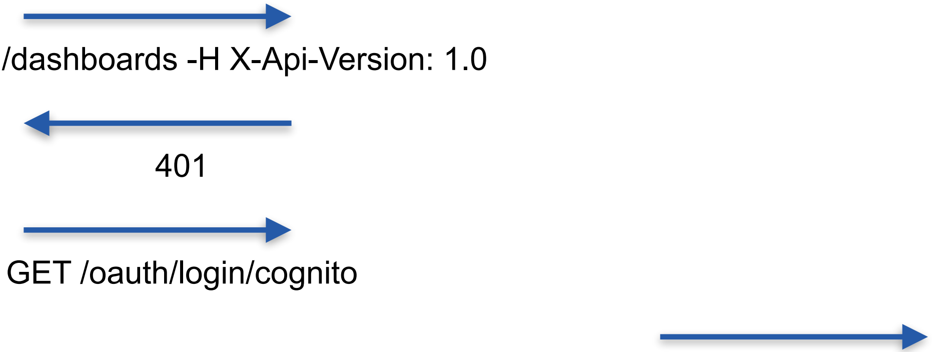
401

GET /oauth/login/cognito

dashboard.clientabc.com

api.micronautdashboard.example

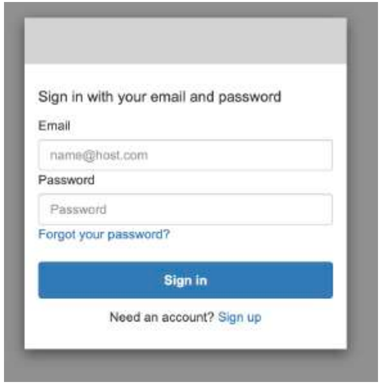
auth.micronautdashboard.example



GET /api/dashboards -H X-Api-Version: 1.0

401

GET /oauth/login/cognito



Sign in with your email and password

Email

name@host.com

Password

Password

Forgot your password?

Need an account? Sign up

GET /login?scope=openid+email+profile&response_type=code&redirect_uri=...

dashboard.clientabc.com

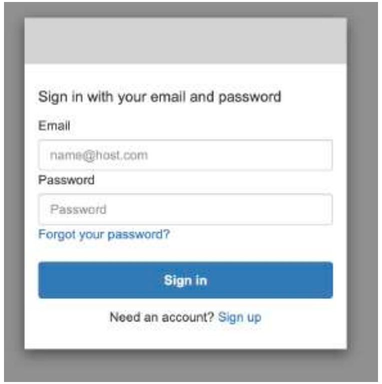
api.micronautdashboard.example

auth.micronautdashboard.example

GET /api/dashboards -H X-Api-Version: 1.0

401

GET /oauth/login/cognito



Sign in with your email and password

Email

name@host.com

Password

Password

Forgot your password?

Sign in

Need an account? Sign up

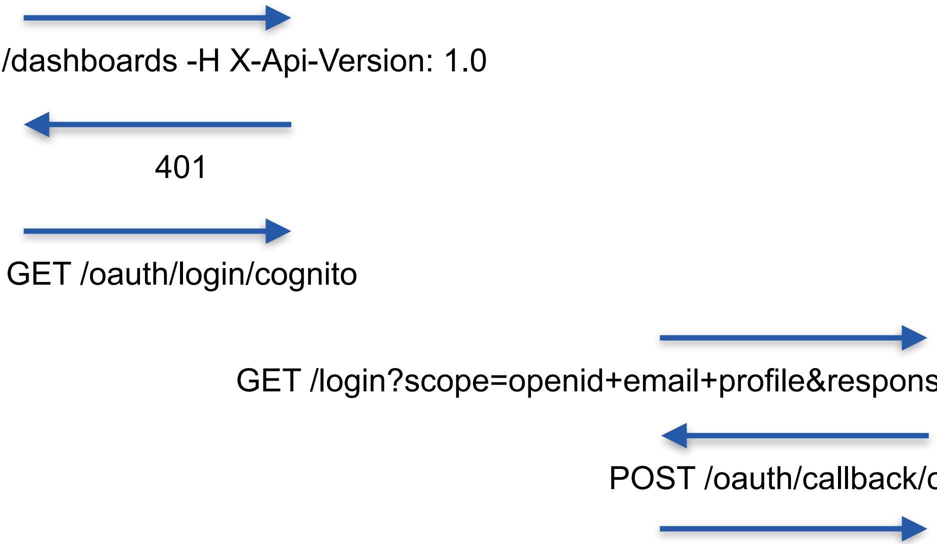
GET /login?scope=openid+email+profile&response_type=code&redirect_uri=...

POST /oauth/callback/cognito

dashboard.clientabc.com

api.micronautdashboard.example

auth.micronautdashboard.example



GET /api/dashboards -H X-Api-Version: 1.0

401

GET /oauth/login/cognito

GET /login?scope=openid+email+profile&response_type=code&redirect_uri=...

POST /oauth/callback/cognito

POST /oauth2/token

Sign in with your email and password

Email

name@host.com

Password

Password

Forgot your password?

Need an account? Sign up

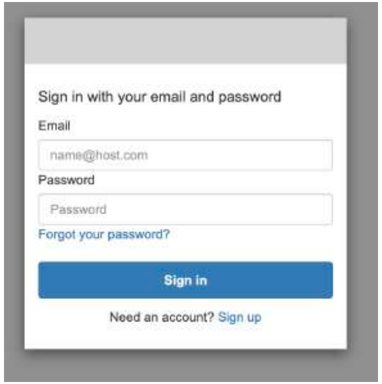
dashboard.clientabc.com | api.micronautdashboard.example | auth.micronautdashboard.example

→ GET /api/dashboards -H X-Api-Version: 1.0

← 401

→ GET /oauth/login/cognito

→ GET /login?scope=openid+email+profile&response_type=code&redirect_uri=...



Sign in with your email and password

Email

name@host.com

Password

Password

Forgot your password?

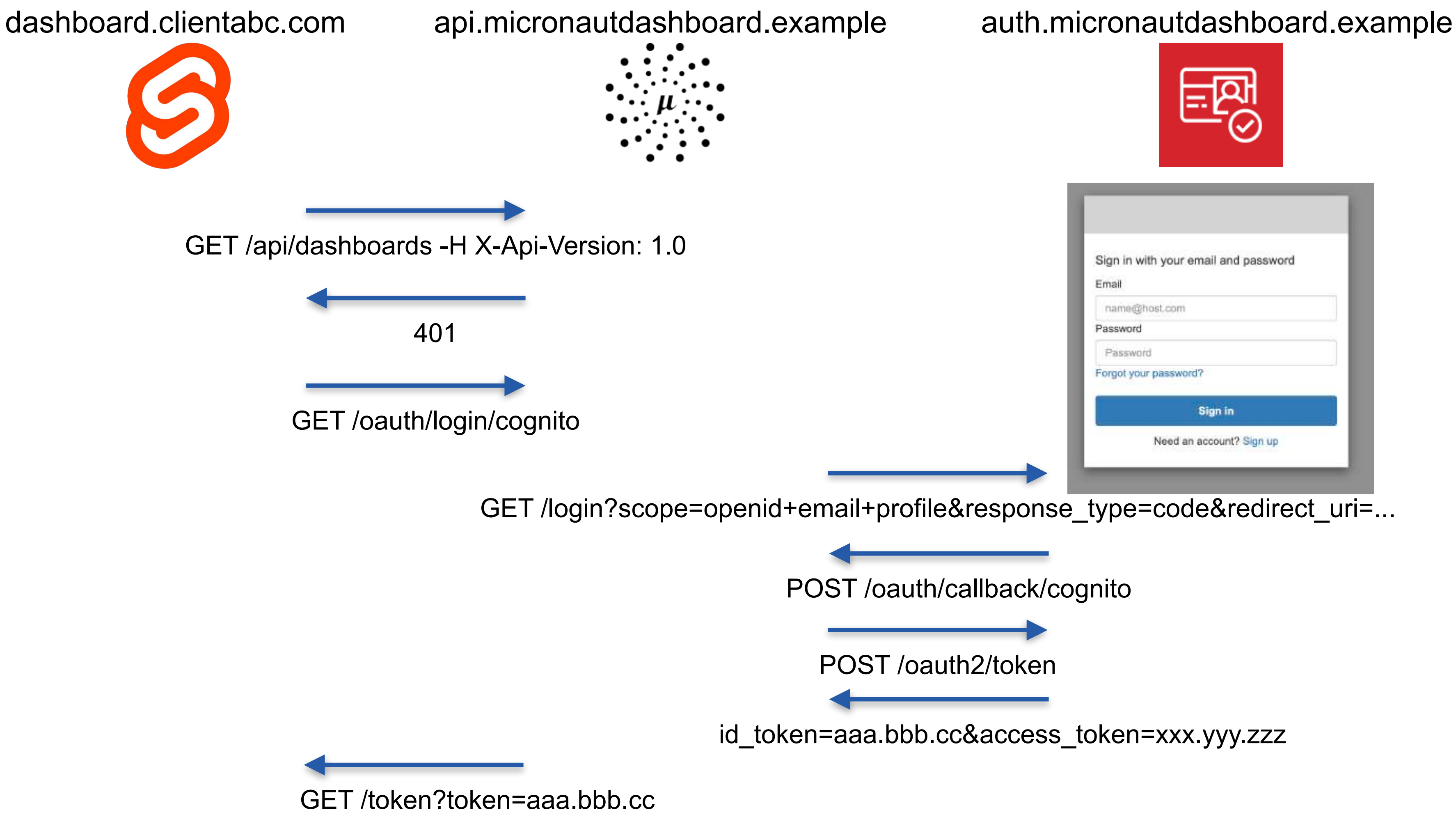
Sign in

Need an account? Sign up

← POST /oauth/callback/cognito

→ POST /oauth2/token

← id_token=aaa.bbb.cc&access_token=xxx.yyy.zzz



dashboard.clientabc.com

api.micronautdashboard.example

auth.micronautdashboard.example

GET /api/dashboards -H X-Api-Version: 1.0

401

GET /oauth/login/cognito

GET /login?scope=openid+email+profile&response_type=code&redirect_uri=...

POST /oauth/callback/cognito

POST /oauth2/token

id_token=aaa.bbb.cc&access_token=xxx.yyy.zzz

GET /token?token=aaa.bbb.cc

Sign in with your email and password

Email

name@host.com

Password

Password

Forgot your password?

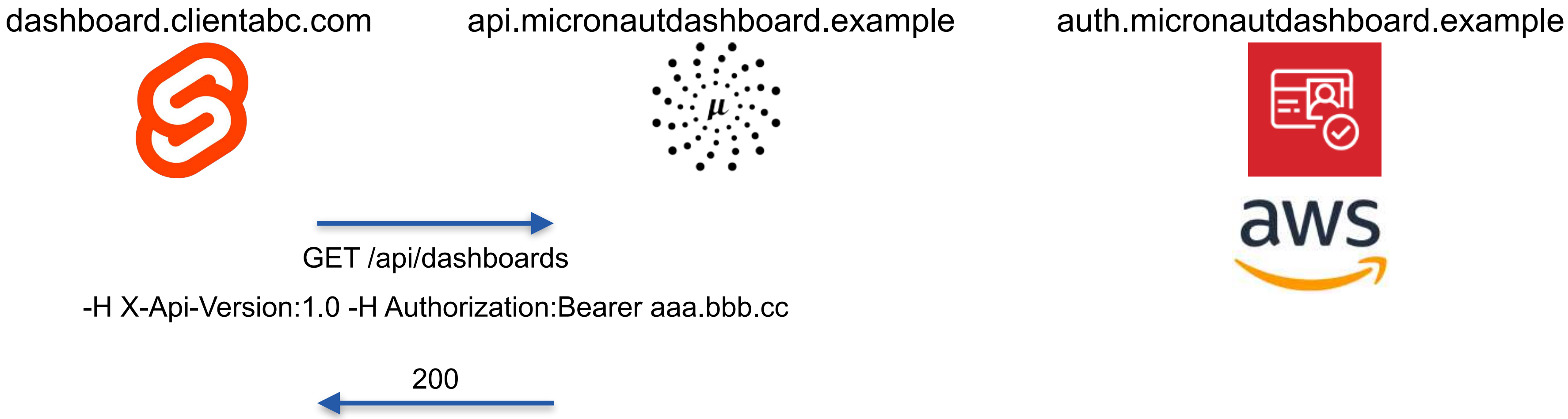
Sign in

Need an account? Sign up

Bearer Token

OBJECT COMPUTING

HOME TO MICRONAUT®



dashboard.clientabc.com

api.micronautdashboard.example

auth.micronautdashboard.example

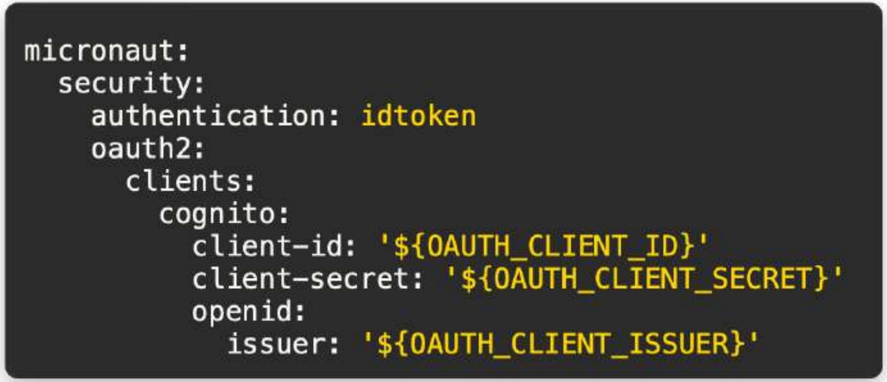
GET /api/dashboards

-H X-Api-Version:1.0 -H Authorization:Bearer aaa.bbb.cc

200

OAuth 2.0 - OpenID Connect

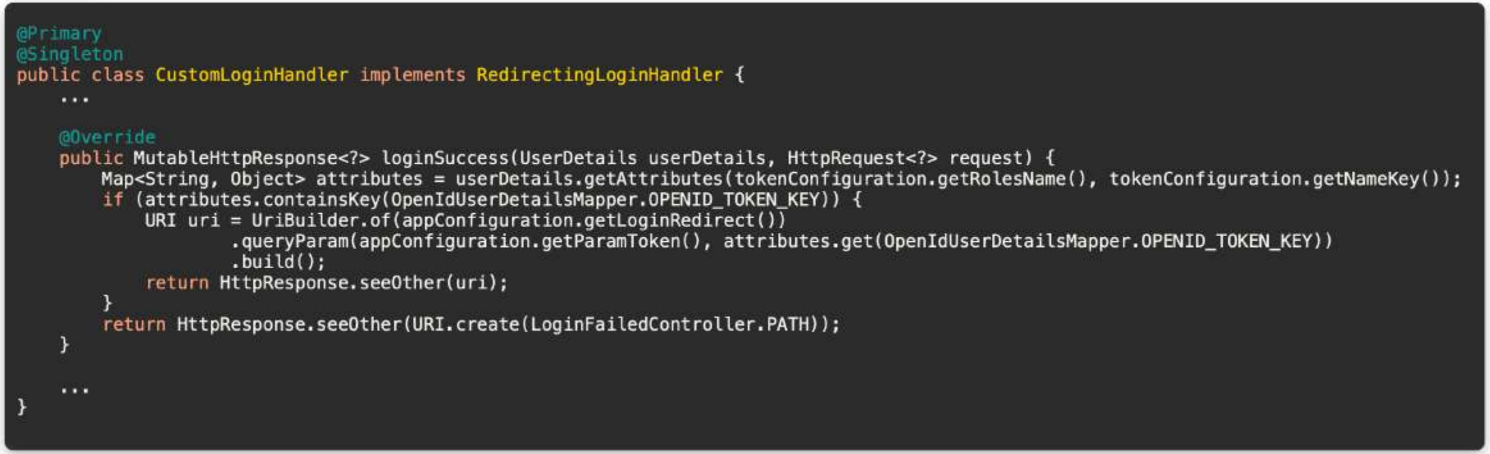
src/main/resources/application.yml



```
micronaut:
  security:
    authentication: idtoken
    oauth2:
      clients:
        cognito:
          client-id: '${OAUTH_CLIENT_ID}'
          client-secret: '${OAUTH_CLIENT_SECRET}'
          openid:
            issuer: '${OAUTH_CLIENT_ISSUER}'
```

OAuth 2.0 - OpenID Connect

src/main/resources/application.yml

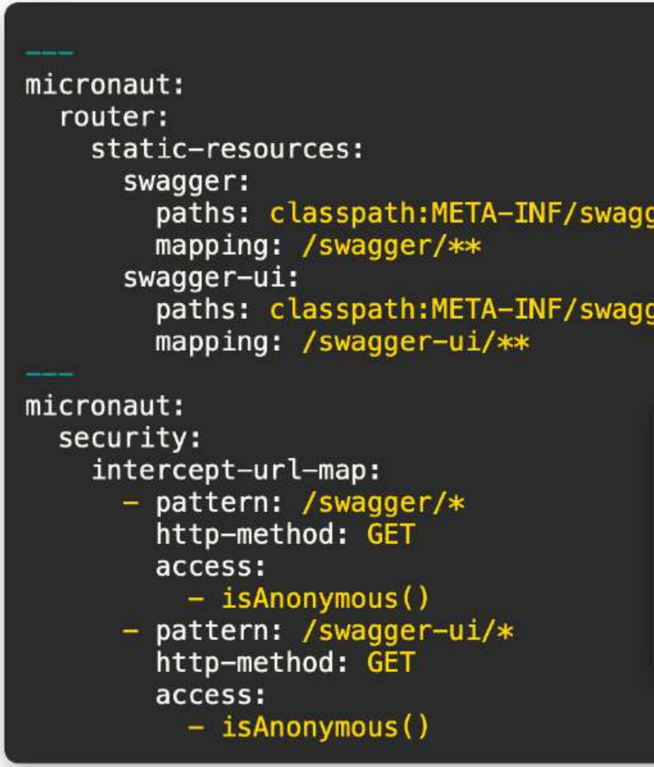


```
@Primary
@Singleton
public class CustomLoginHandler implements RedirectingLoginHandler {
    ...

    @Override
    public MutableHttpResponse<?> loginSuccess(UserDetails userDetails, HttpRequest<?> request) {
        Map<String, Object> attributes = userDetails.getAttributes(tokenConfiguration.getRolesName(), tokenConfiguration.getNameKey());
        if (attributes.containsKey(OpenIdUserDetailsMapper.OPENID_TOKEN_KEY)) {
            URI uri = UriBuilder.of(appConfiguration.getLoginRedirect())
                    .queryParam(appConfiguration.getParamToken(), attributes.get(OpenIdUserDetailsMapper.OPENID_TOKEN_KEY))
                    .build();
            return HttpResponse.seeOther(uri);
        }
        return HttpResponse.seeOther(URI.create(LoginFailedController.PATH));
    }

    ...
}
```

OPENAPI
INITIATIVE

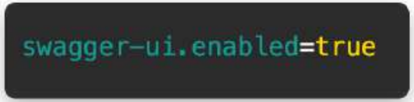
API Documentation - Swagger UI

src/main/resources/application.yml



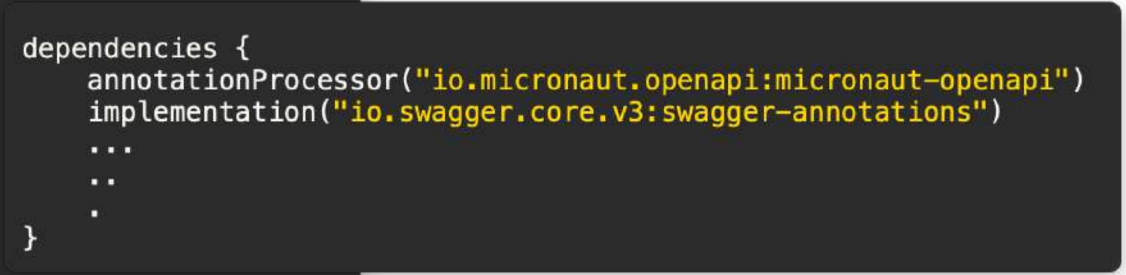
```
---
micronaut:
  router:
    static-resources:
      swagger:
        paths: classpath:META-INF/swagger
        mapping: /swagger/**
      swagger-ui:
        paths: classpath:META-INF/swagger/views/swagger-ui
        mapping: /swagger-ui/**
---
micronaut:
  security:
    intercept-url-map:
      - pattern: /swagger/*
        http-method: GET
        access:
          - isAnonymous()
      - pattern: /swagger-ui/*
        http-method: GET
        access:
          - isAnonymous()
```

openapi.properties

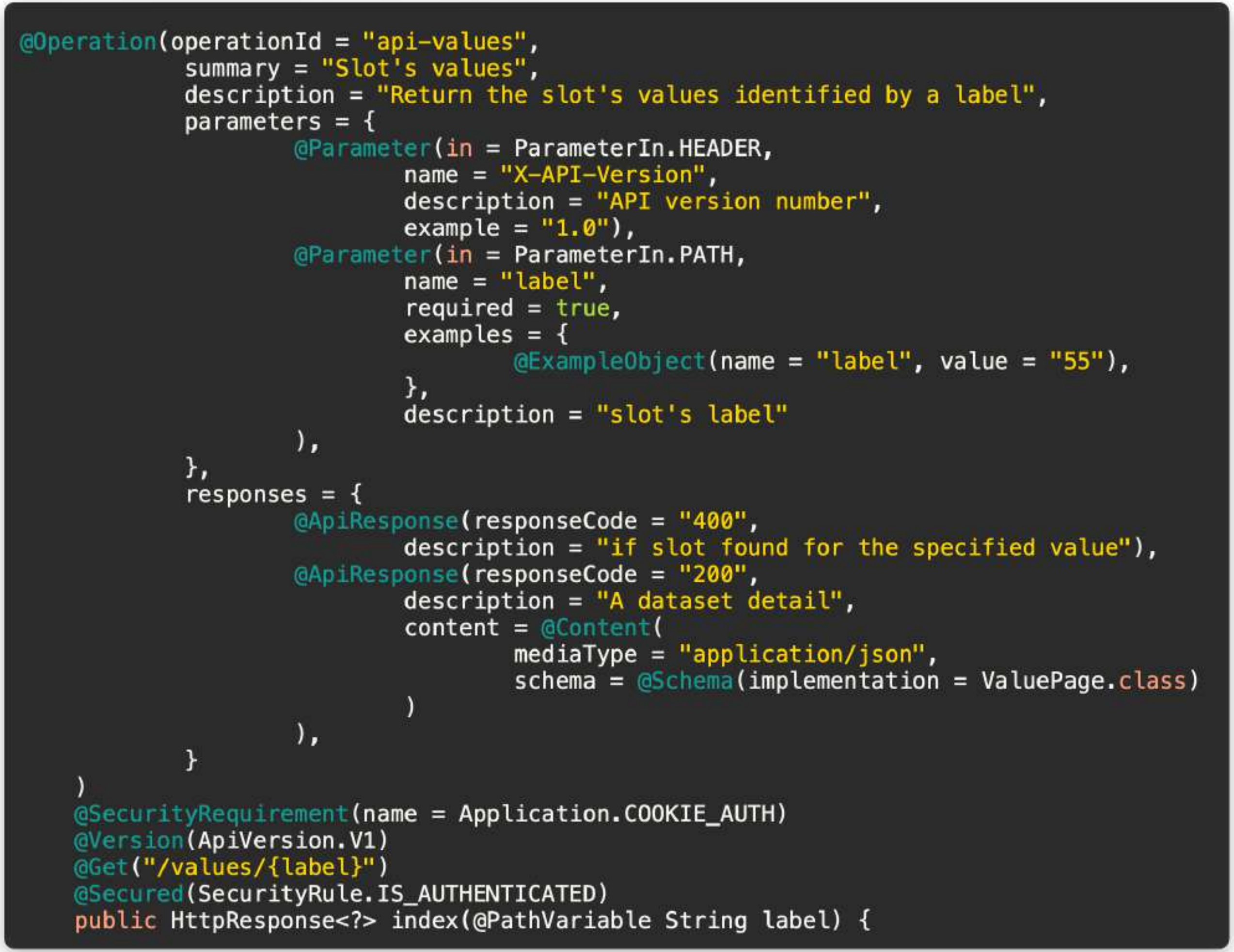


```
swagger-ui.enabled=true
```

build.gradle



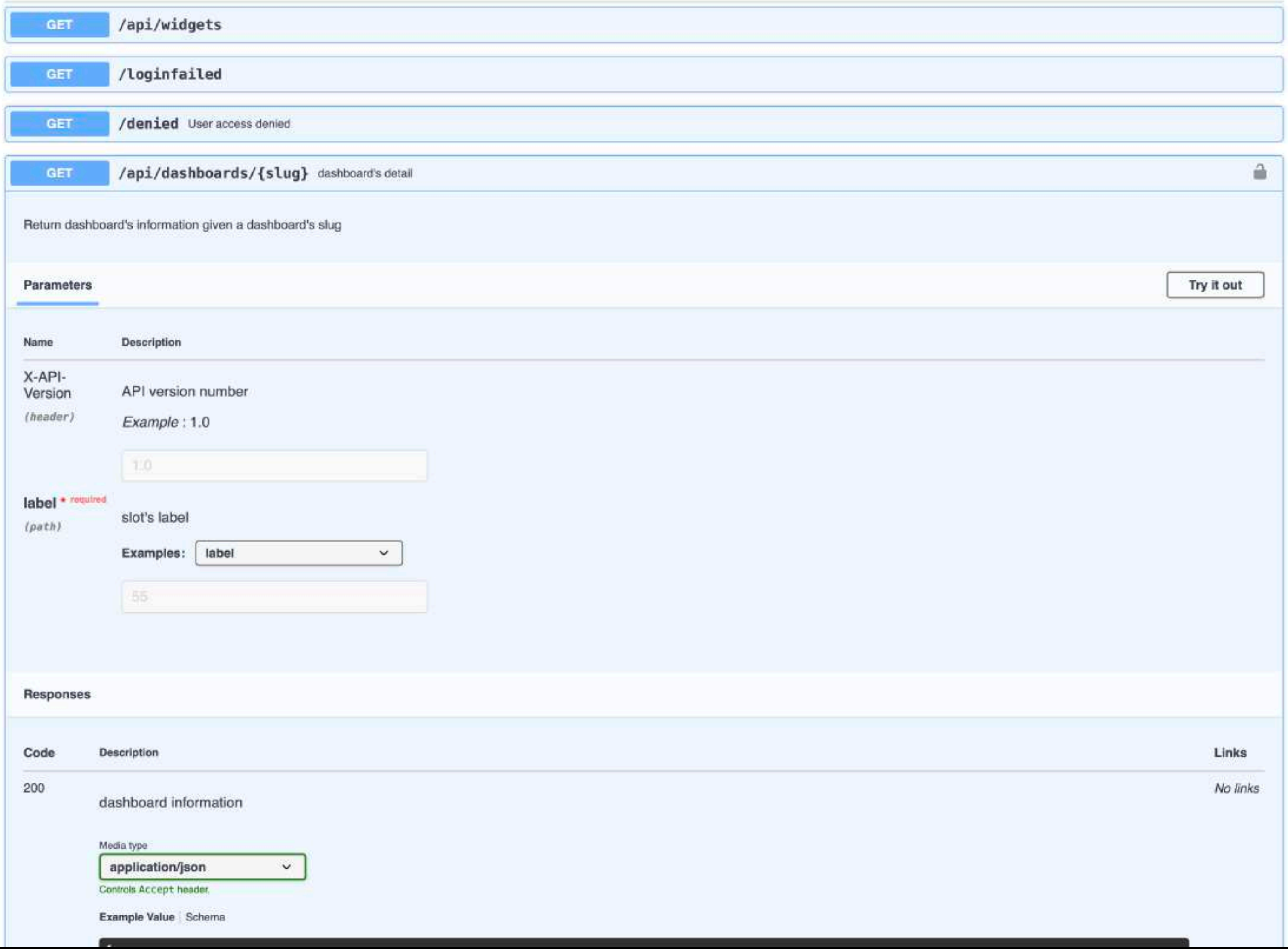
```
dependencies {
    annotationProcessor("io.micronaut.openapi:micronaut-openapi")
    implementation("io.swagger.core.v3:swagger-annotations")
    ...
    ..
    .
}
```



```
@Operation(operationId = "api-values",
            summary = "Slot's values",
            description = "Return the slot's values identified by a label",
            parameters = {
                    @Parameter(in = ParameterIn.HEADER,
                            name = "X-API-Version",
                            description = "API version number",
                            example = "1.0"),
                    @Parameter(in = ParameterIn.PATH,
                            name = "label",
                            required = true,
                            examples = {
                                    @ExampleObject(name = "label", value = "55"),
                            },
                            description = "slot's label"
                    ),
            },
            responses = {
                    @ApiResponse(responseCode = "400",
                            description = "if slot found for the specified value"),
                    @ApiResponse(responseCode = "200",
                            description = "A dataset detail",
                            content = @Content(
                                    mediaType = "application/json",
                                    schema = @Schema(implementation = ValuePage.class)
                            )
                    ),
            }
    )
    @SecurityRequirement(name = Application.COOKIE_AUTH)
    @Version(ApiVersion.V1)
    @Get("/values/{label}")
    @Secured(SecurityRule.IS_AUTHENTICATED)
    public HttpResponse<?> index(@PathVariable String label) {
```

OBJECT COMPUTING

HOME TO MICRONAUT®



/api/widgets

/loginfailed

/denied User access denied

/api/dashboards/{slug} dashboard's detail

Return dashboard's information given a dashboard's slug

Parameters

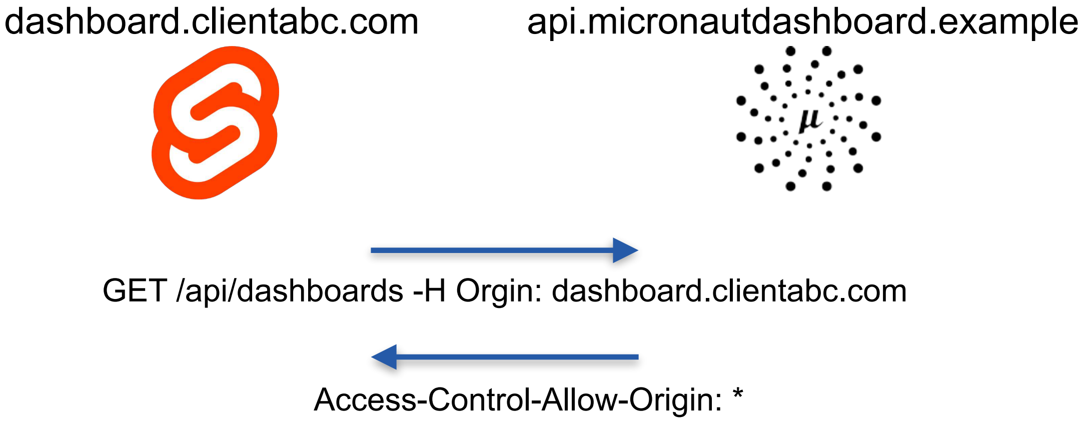
Name	Description
X-API-Version *(header)*	API version number *Example* : 1.0 1.0
label * required *(path)*	slot's label **Examples:** label 55

Responses

Code	Description	Links
200	dashboard information Media type application/json Controls Accept header. **Example Value** | Schema	*No links*

CORS

CORS - Cross Origin Resource Sharing



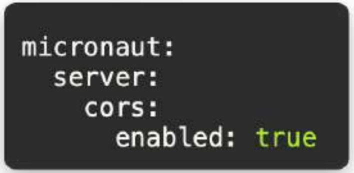
dashboard.clientabc.com

api.micronautdashboard.example

GET /api/dashboards -H Orgin: dashboard.clientabc.com

Access-Control-Allow-Origin: *

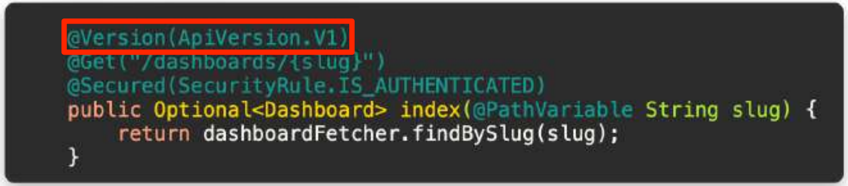
src/main/resources/application.yml



```
micronaut:
  server:
    cors:
      enabled: true
```

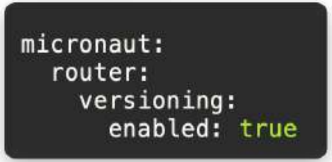
Versioning

Versioning



```java
@Version(ApiVersion.V1)
@Get("/dashboards/{slug}")
@Secured(SecurityRule.IS_AUTHENTICATED)
public Optional<Dashboard> index(@PathVariable String slug) {
    return dashboardFetcher.findBySlug(slug);
}
```

src/main/resources/application.yml



```yaml
micronaut:
  router:
    versioning:
      enabled: true
```

OBJECT COMPUTING
HOME TO MICRONAUT®

What is Svelte?

- Alternative to web frameworks like React, Vue, and Angular
- A web application **compiler**, not a runtime library
  - implemented in TypeScript
  - compiles **.svelte** files to a single Javascript file
  - no Svelte runtime dependencies
- Doesn’t use a virtual DOM
- Develop by **Rich Harris**
  - formerly at “The Guardian”; currently at “The New York Times”
  - previously created **Ractive** web framework - https://ractive.js.org/
    - used at “The Guardian”
    - inspired parts of Vue
- created **Rollup** module bundler - https://rollupjs.org/
  - alternative to Webpack and Parcel

Top Svelte Features

- Requires less code
- Small bundle sizes
- File-based component definitions
- CSS scoped by default
- Clear place to put global CSS
  - `public/global.css`
- Two-way data bindings
  - between top-level variables and form elements or components
- Supports TypeScript

- Easy state management within components
  - using top-level variables and reactivity
  - using reactive statements - `$:`
- Easy state management across components
  - using stores which are an alternative to React Redux, Vue Vuex, and Angular NgRx
- Built-in support for animations
  - https://www.youtube.com/watch?v=wMpKD0IDiHc
- Can compile Svelte components to custom elements (web components)

Starting a New Svelte App

- Steps
  - `npx degit sveltejs/template app-name`
  - `cd app-name`
  - `npm install`
  - `npm run dev`
- Use same tools as for other kinds of web apps
  - ESLint, Prettier, Jest, svelte-testing-library, Cypress, Storybook, ...

Svelte Component Source Files

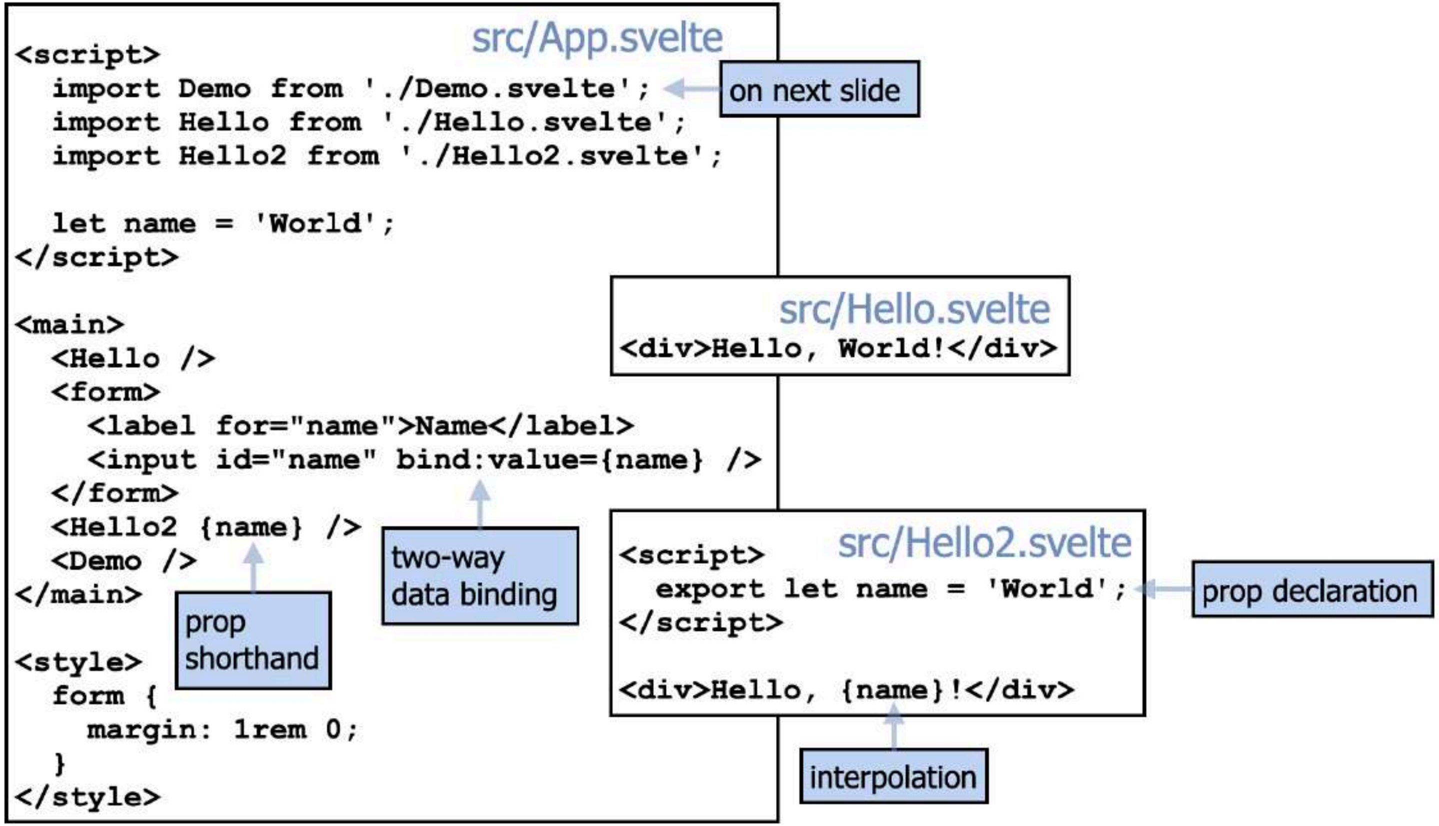
OBJECT COMPUTING
HOME TO MICRONAUT®

src/ComponentName.svelte

```
<script>
  // imports go here
  // prop declarations go here
</script>

<!-- HTML goes here, including
     interpolations and special block syntax. -->

<style>
  /* CSS goes here */
</style>
```

`script` and `style` elements are optional

Example components



src/App.svelte

```
<script>
  import Demo from './Demo.svelte';
  import Hello from './Hello.svelte';
  import Hello2 from './Hello2.svelte';

  let name = 'World';
</script>

<main>
  <Hello />
  <form>
    <label for="name">Name</label>
    <input id="name" bind:value={name} />
  </form>
  <Hello2 {name} />
  <Demo />
</main>

<style>
  form {
    margin: 1rem 0;
  }
</style>
```

on next slide

prop shorthand

two-way data binding

src/Hello.svelte

```
<div>Hello, World!</div>
```

src/Hello2.svelte

```
<script>
  export let name = 'World';
</script>

<div>Hello, {name}!</div>
```

prop declaration

interpolation

src/Demo.svelte

```
<script>
  let count = 0;
  $: tooMany = count >= 10;
</script>

<button class:danger={tooMany} on:click={() => count++}>
  Pressed {count} times
</button>
{#if tooMany}
  <p>You have pressed the button too many times!</p>
{/if}

<style>
  button {
    border: solid gray 1px;
    border-radius: 4px;
    margin-top: 1rem;
    padding: 0.5rem;
  }

  .danger {
    background-color: red;
    color: white;
  }
</style>
```

top-level variable

reactive statement

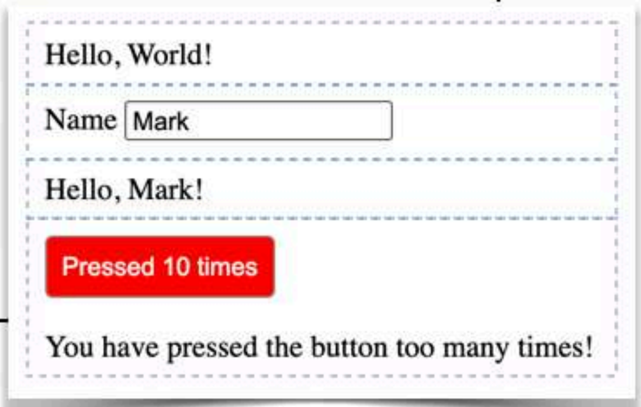
event handling

interpolation

Mustache-style markup

styles are scoped to this component

Top-level variables ARE the **component state** if used in HTML! When state changes, only the relevant parts of DOM are updated.



Hello, World!

Name Mark

Hello, Mark!

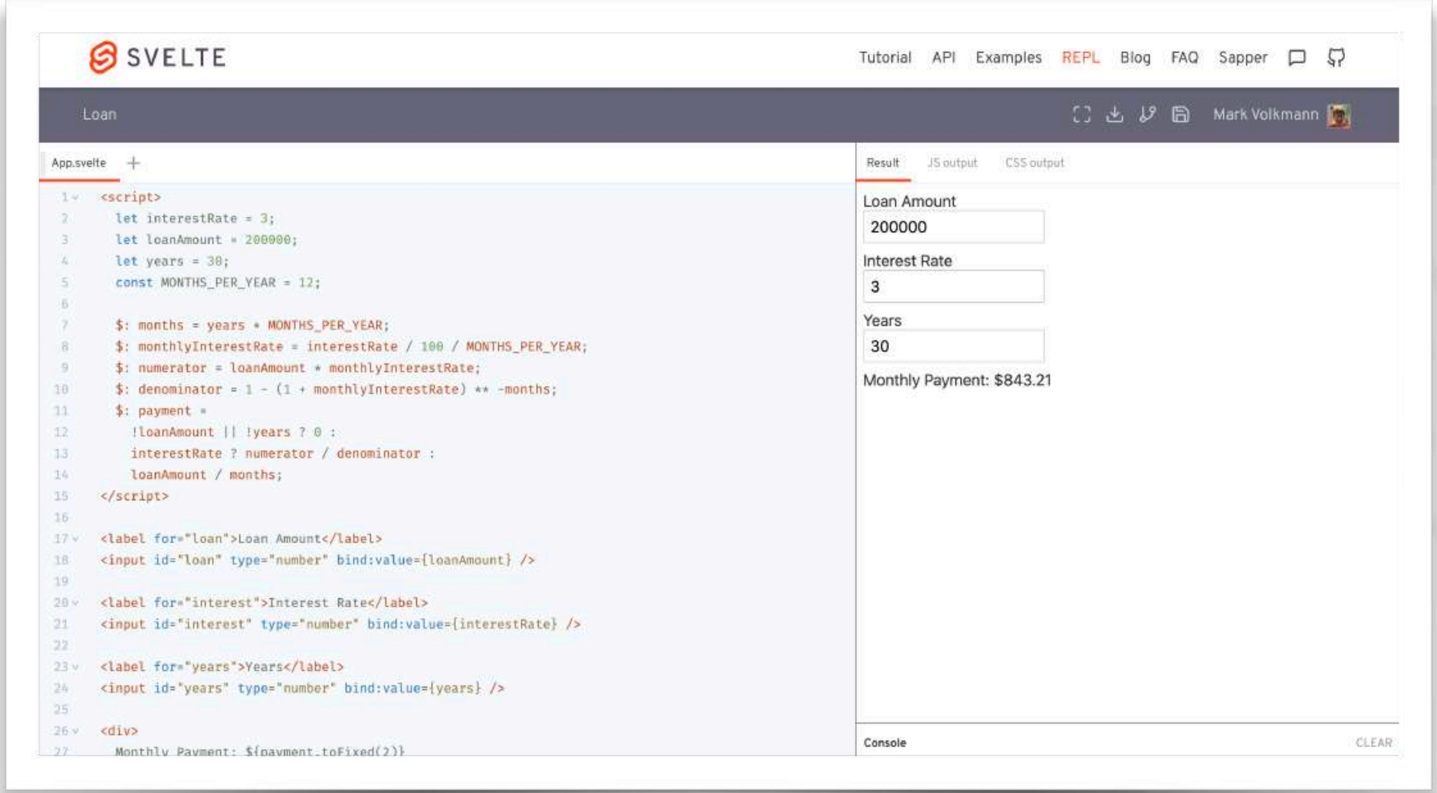
You have pressed the button too many times!

Svelte REPL

- Online tool for experimenting with Svelte
  - https://svelte.dev/repl/
  - don’t need to install anything
  - can download apps to continue development locally
- Login to GitHub account to enable saving and recalling named apps
- Select from list of saved apps to recall them
- Share unique app URLs with others
- Loan Calculator
  - great example of using reactive declarations and two way bindings
  - see https://tinyurl.com/svelte-loan and the next slide

Svelte REPL

OBJECT COMPUTING
HOME TO MICRONAUT®



```
<script>
  let interestRate = 3;
  let loanAmount = 200000;
  let years = 30;
  const MONTHS_PER_YEAR = 12;

  $: months = years * MONTHS_PER_YEAR;
  $: monthlyInterestRate = interestRate / 100 / MONTHS_PER_YEAR;
  $: numerator = loanAmount * monthlyInterestRate;
  $: denominator = 1 - (1 + monthlyInterestRate) ** -months;
  $: payment =
    !loanAmount || !years ? 0 :
    interestRate ? numerator / denominator :
    loanAmount / months;
</script>

<label for="loan">Loan Amount</label>
<input id="loan" type="number" bind:value={loanAmount} />

<label for="interest">Interest Rate</label>
<input id="interest" type="number" bind:value={interestRate} />

<label for="years">Years</label>
<input id="years" type="number" bind:value={years} />

<div>
  Monthly Payment: ${payment.toFixed(2)}
```

Loan Amount
200000

Interest Rate
3

Years
30

Monthly Payment: $843.21

Kind of Stores

Our demo app doesn’t uses stores,
but they are an important feature of Svelte
and warrant some discussion.

- **Writable**
  - only kind that can be modified by components
  - methods
    - `set(newValue)`
    - `update(currentValue => newValue)`

calculates new value from current value

- **Readable**
  - handle computing their data, perhaps from a REST call
  - components cannot modify

- **Derived**
  - derive data from current values of other stores

- **Custom**
  - must implement `subscribe` method
  - can provide custom methods to update state and not expose `set` and `update` methods

Defining Writable Stores

src/stores.js

```
import {writable} from 'svelte/store';

export const dogStore = writable([]);
```

initial value

using optional second argument

```
export const fancyStore = writable(
  initialValue,
  async set => {
    // Called when subscriber count goes from 0 to 1.
    // Compute initial value and pass to set function.
    const res = await fetch('/some/url');
    const data = await res.json();
    set(data);

    return () => {
      // Called when subscriber count goes to 0.
    }
  }
);
```

Using Stores

- Option #1 - **subscribe** method - very verbose!
- Option #2 - **$** auto-subscription shorthand - much better!
  - variables whose names begin with **$** must be stores
  - automatically subscribes when first used
    and unsubscribes when removed from DOM

uses **subscribe** method

```
<script>
  import {onDestroy} from 'svelte';
  import {dogStore} from './stores.js';
  let dogs;
  const unsubscribe = dogStore.subscribe(value => dogs = value);
  onDestroy(unsubscribe);
</script>

<!-- Use dogs in HTML. -->
```

uses auto-subscription

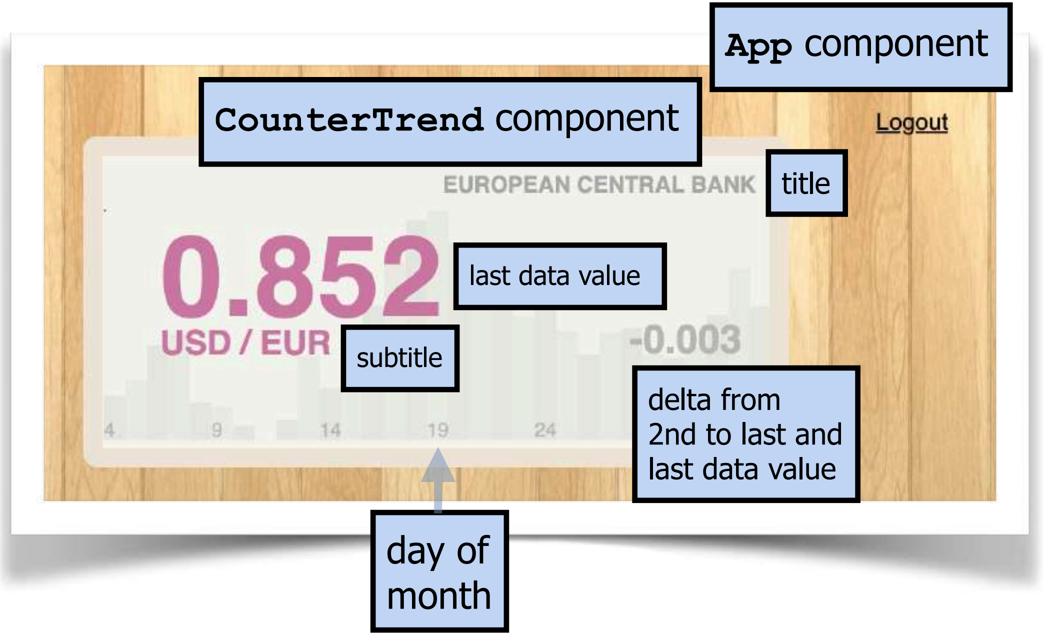
```
<script>
  import {dogStore} from './stores.js';
</script>

<!-- Use $dogStore in HTML. -->
```

Related Tools

- **Svelte VS Code extension**
- **Sapper** - https://sapper.svelte.dev/
  - “application framework powered by Svelte”
  - similar to Next and Gatsby
  - provides routing, server-side rendering, code splitting, and building static sites
- **Svelte Native** - https://svelte-native.technology/
  - for implementing native mobile apps
  - based on NativeScript
  - community-driven project
- **Svelte Testing Library** - https://testing-library.com/docs/svelte-testing-library/intro
- **Storybook** with Svelte - https://storybook.js.org/docs/guides/guide-svelte/

Svelte Resources

- **“Rethinking Reactivity” talk by Rich Harris**
  - delivered multiple times, most recently at “Shift Conference” June 20, 2019
  - explains issues with using virtual DOM (like React and Vue) and motivation for Svelte
- **Home page** - https://svelte.dev
  - contains **Tutorial**, **API Docs**, **Examples**, online **REPL**, **Blog**, and **Sapper** link
    - REPL is great for trying small amounts of Svelte code
    - REPL can save for sharing and submitting issues
- **Discord chat room** - https://discordapp.com/invite/yy75DKs
- **GitHub** - https://github.com/sveltejs/svelte
- **Awesome Svelte** - https://github.com/CalvinWalzel/awesome-svelte
- **Awesome Svelte Resources** - https://github.com/ryanatkn/awesome-svelte-resources



App component

CounterTrend component

title

last data value

subtitle

delta from 2nd to last and last data value

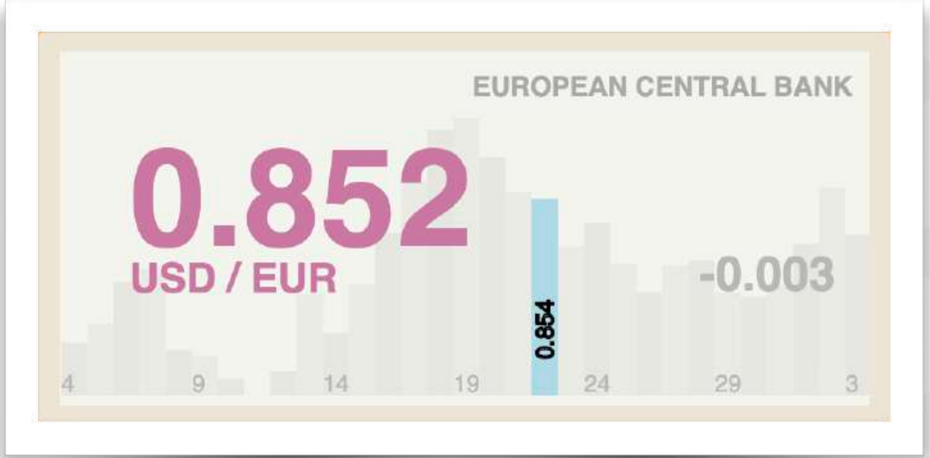
day of month

… Svelte and Micronaut Sample App

- Svelte app looks for token in query parameter or `sessionStorage`
- If no token, calls `/oauth/ogin/cognito` function
- If token, calls `loadData` function and passes data to `CounterTrend` component for rendering
- `loadData` function uses Fetch API to send HTTP requests to two REST services, both of which require `Authorization` header set to "`Bearer {token}`"
- Svelte `login` function just redirects to Cognito login URL
- After successful login, Cognito login page redirects back to Svelte app, including `token` query parameter
- Svelte app stores token in `sessionStorage`
- Svelte `logout` function removes token from `sessionStorage` and redirects to logout URL when renders login page again

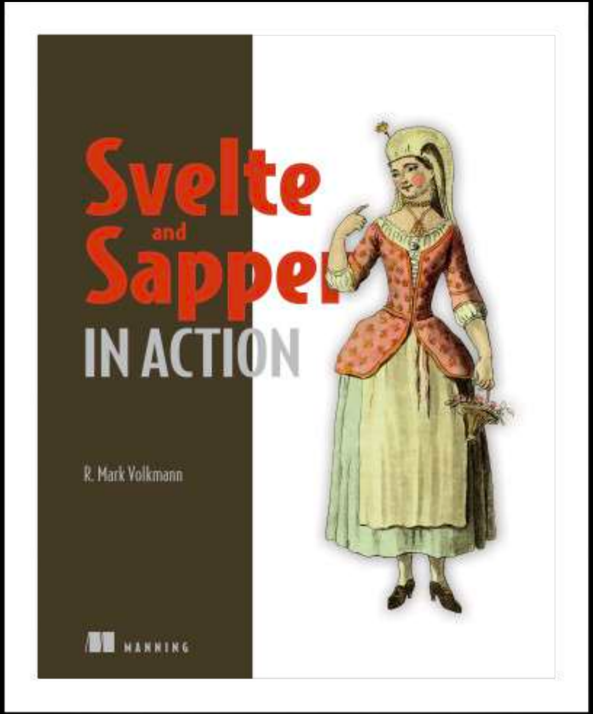
CounterTrend Component

- Uses SVG `<rect>` and `<text>` elements to create a bar chart
- Hover over a bar to see its value
- In this demo it is used to show the change in exchange rate between USD and Euros



OBJECT COMPUTING
HOME TO MICRONAUT®

- **Svelte is a worthy alternative** to the current popular options of React, Vue, and Angular
- Check out my book, “Svelte and Sapper in Action”
  - https://www.manning.com/books/svelte-and-sapper-in-action



GRAILS® & GROOVY COMMUNITY RESOURCES

- @grailsframework
- Grails.org
- github.com/grails
- Grails-slack.cfapps.io
- grails.org/blog/index.html
- grails.org/foundation
- objectcomputing.com/products/grails/solutions
- objectcomputing.com/products/grails
- @ApacheGroovy
- groovy-lang.org
- github.com/apache/groovy
- groovy-community.slack.com
- Apache.org

Q&A

?

MICRONAUT® COMMUNITY RESOURCES

- @micronautfw
- gitter.im/micronautfw
- micronaut.io/launch
- docs.micronaut.io
- micronaut.io/learn
- guides.micronaut.io
- micronaut.io/blog
- micronaut.io/faq
- micronaut.io/foundation
- github.com/micronaut-projects/micronaut-core
- objectcomputing.com/products/micronaut/solutions
- objectcomputing.com/products/micronaut
- objectcomputing.com/resources/events

OBJECT COMPUTING
HOME TO MICRONAUT®

THANK YOU!

LET'S STAY CONNECTED

OBJECT COMPUTING
HOME TO MICRONAUT®

✉ info@micronaut.io

✉ 2GM@objectcomputing.com

@micronautfw

@grailsframework

@ApacheGroovy