worldskills Europe

EuroSkills Test Project

Web Development 17

Module B

Module B – Dynamic website with server-side rendering

The goal of Module B is to create a server-side rendered website which customers can use to manage their API usage and billing. It is possible to use additional libraries in the frontend for single interactive elements, but the main part of the website (e.g. lists and detail views) must be rendered server-side and therefore also work when JavaScript is disabled in the browser.

The project does not have a database yet, and it is therefore also in the scope of this task to come up with a new database design and import a provided CSV file with partial example data.

As this website will be publicly exposed, it must implement the OWASP guidelines.

Competitor Information

Module B will be assessed using the provided version of Google Chrome. Different security aspects will be tested.

The design of the website is not important in this first iteration. The client will mostly focus on the functionality, but some basic styling is expected to make it readable and usable.

Website Requirements

The website must provide the following functionality.

Login

All other pages are protected and not accessible to non-authenticated users. Login must be possible by providing a username and a password. As the first version of the website will only allow users to sign up by invitation only, it is not necessary to be able to register accounts.

However, please create the following accounts:

- Username: demo1
  Password: skills2023d1
- Username: demo2
  Password: skills2023d2

The password must be stored in a secure way (hashed) in case someone gets access to the database.

Workspaces

Users can create as many workspaces as they like. Workspaces act as a way to separate the API usage. All of the following functionality (API tokens, billing quotas, bills) are scoped to a workspace.

After login, the user is redirected to their list of workspaces. On that page, they can create or update workspaces. Users can only access and modify their own workspaces.

For each workspace, they have the possibility to manage the API tokens, billing quotas, and bills. This additional functionality can also be provided on separate pages through links and does not have to be on the same page.

A workspace has the following attributes:

- A required **title** (max 100 characters, unique within the user account)
- An optional **description** of any length

API Tokens

For each workspace, it is possible to create one or more API tokens. All available tokens of a workspace are listed with their name and the creation date.

The actual token is only revealed once when it is created. It is not possible to view the token again after creation.

Each token can be revoked. If it is revoked, it cannot be used anymore. It is also not possible to activate a revoked token again. In the token list, it is clearly visible when a token is revoked and the revocation date is shown.

A token has the following attributes:

- A required **name** (max 100 characters)
- A randomly generated **token** of at least 40 characters
- An automatically set **creation date**
- A **revocation date** which is set once it is revoked

Billing Quotas

It is possible to set and remove a user-defined billing quota per workspace. Each call to the provided API costs a certain amount of money. If a billing quota is set, it defines the maximum amount that can be spent on API calls within the assigned workspace per calendar month. If the quota is exceeded, usage of the API is not possible anymore for all API tokens of this workspace.

Quotas are displayed in the following way:

- If no quota is set, the costs of the current calendar month is shown, but it must also be clear that there is no maximum.
- If a quota is set, the costs and the maximum of the current calendar month are shown. Also, the number of remaining days in the current billing cycle is shown so users know when it will reset.

A billing quota has the following attributes:

- A **limit** in dollars for each calendar month

Bills

For each passed calendar month, a bill is generated and visible to the users.

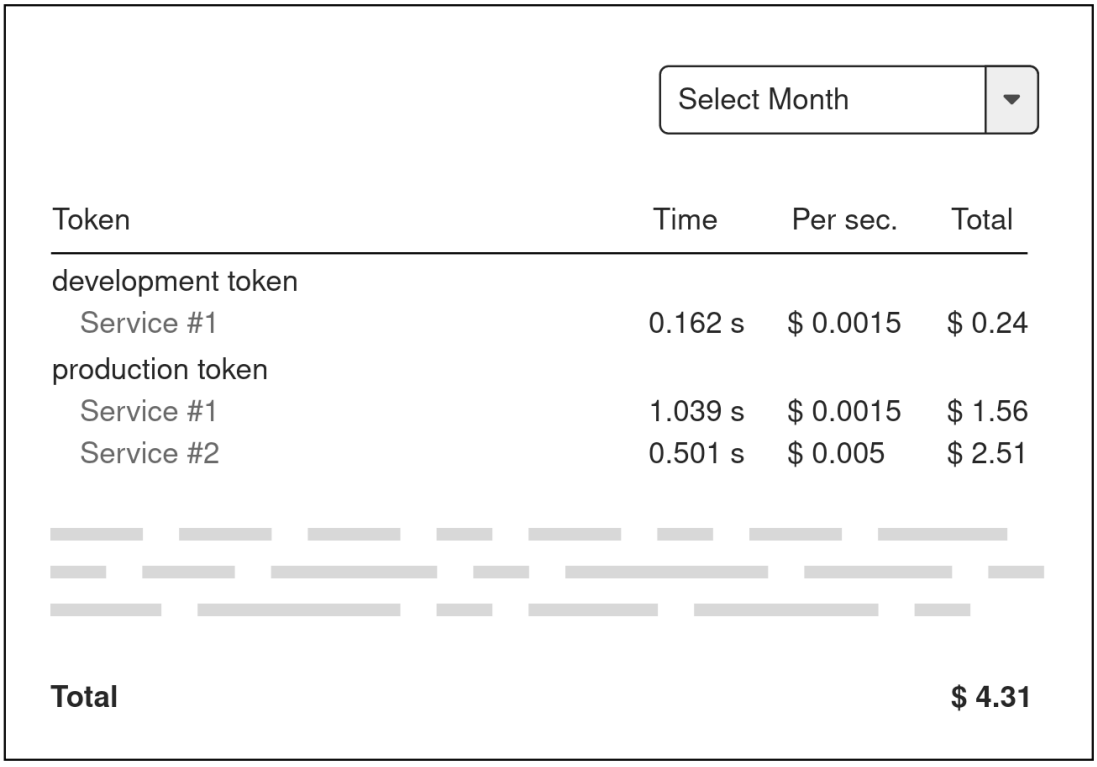
Each API token can access services that are exposed through an API. The user pays for using those services based on the time it takes to compute the result. Each service can cost a different amount of money per second.

The bill contains the following data per API token and per accessed service which is exposed through an API:

- Usage of the service for a specific API token in seconds
- Cost of the service per second (static value per service)
- Usage cost of the specific token and service
- If a token did not access a service, that service must not be listed
- If a token was not used at all, that token must not be listed

A total row will also show the total cost over all API tokens and services. The total costs are rounded to two decimal points for displaying purpose only.

The client has provided the following example mockup to show their idea of a bill. However, it is possible to change it or come up with a completely different layout.



Example Data

To already have some example data to generate bills and check if calculations are correct, the client provided a CSV file with some billing related example data.

Import the data of this file into your own database schema. The imported data must be normalized.