

Lab 2 - Callbacks/Async/Promises

Helpful material for this lab:

At least one of...

- [Promise \(Mozilla\)](#)
- [Promise \(Google\)](#)
- [Promise \(JS Info\)](#)

And...

- [axios cheet sheet](#)

Task

Allright, let's get to it! You have been given the files `index.html` and `style.css` for you to do your magic on. Open the HTML document in a browser and you should see an anonymous dog, an anonymous person and a form asking for some odd info. There is not really much context behind any of them, it's just an assortment of weird tasks :)

Task 1 - Who's a good boy?

A customer who loves dogs want you to deliver random pictures of dogs to display in a dog conference (whatever that is). Being a person with a lot of street cred, you call your local web service dealer hook you up with a fresh dog web service API. The result is [Dog CEO](#) which supplies you with 100% pure, uncut images of dogs.

Your task is as following:

- When you click the 'Generate dog' button a picture of a random dog should appear instead of the anonymous dog.

Requirements

- Each subsequent click makes a new random dog appear and it should replace the previous one.
- Requests are made to Dog CEO API.
- Either fetch or axios is used to make the requests.

Task 2 - Aliens

So a second customer comes in, who seems to be obsessed with aliens. This person is convinced that not only do they exist, but that they actually are walking among us and that there is an alien connected to every person. You lose a bit of faith in humanity but hey, money is money. The customer wants you to generate random profiles of random people, and an image of their alien. And also, for some reason, the customer wants to know the country the person is from and see a flag of that country. So naturally you call your dealer who hooks you up with not only a [Random User Generator API](#), but also a [Countries API](#), a [Country Flags API](#) and a [robot/alien/monster image generator API](#)! With these fine products, you can get to work.

Your task:

When you click the 'Generate profile' button the following should happen:

- A picture of a random person should appear and replace the large image of the anonymous person.
- Name Nameson should be replaced with a random name.
- Country should be replaced with a country name. And the country's flag should be shown on the right, replacing the anonymous country image.
- The small image should be replaced with an alien image.

Requirements:

- A request to Random User Generator API is made first.
- Alien image should be generated using the name (full name) received from random user generator.

- All data that is filled should be based on the response from the request made to Random User Generator, either directly or indirectly. So when you use the other APIs that are listed, the requests to those APIs use information provided by the response of the first request to Random User Generator.
- Either fetch or axios is used to make the requests.
- Each subsequent click makes a new random profile appear and it should replace the previous one.

Hint: The request to Random User Generator will supply with a name and an image of a person but it will not supply a country's name or country flag. However, it will supply you with a nationality in the form of a country code, which you can use to make a call to [Rest Countries](#). The country flag API uses a country code as input so that one can be extracted from the nationality.

Task 3 - Enough is enough

After dealing with these slightly odd customers you feel you need a vacation so you decide to prepare a bit and make a small trip planner. It is to be used to plan your budget and you want everything to be correct so you make the form for it automatically validated. For that you're obviously going to need some web services and you already know who to call. Within no time, you have the following web services:

- [Rest Countries](#) (once again)
- [Exchange Rates API](#)

Task:

Fill in the information and have each input validated. When submit is clicked, a `post` request with the information should be sent to [JSON placeholder's](#) "posts" endpoint. (So far we've only used `get` requests)

Requirements:

- Trip names must be at least 3 characters.
- Destination country must be populated by countries received from Rest Countries' "all" endpoint
- Expenses must be a positive number
- Currency must be a valid currency (eg. USD SEK etc.)
- Budget (SEK) must be higher than expenses * exchange rate (chosen currency -> SEK)
- If any of the above are not met, an invalid indicator must appear. Most convenient is to use the `<small>` element below each input to write some text saying why it's invalid.
- Validation of an input should happen when focus has been lost of the input. For example, you write something in an input and then you go to the next. At the moment you leave the first input, validation should happen on the first input.
- Submit button should only be enabled if all input fields are valid. Otherwise it is disabled.
- The data of the `post` request should be a JSON object made from the input from the form and its inputs. To do this you can serialize the form, i.e create an object from the form by using its input values. You can use the function defined in `serialize-form.js` for this.
- Either fetch or axios is used to make the requests.
- The response received from the `post` request should be a JSON object containing the same data that you sent plus an extra `id` property. This can be checked by printing the response to the console.