

Lecture 18: APIs & Frontend

AC215

Shivas Jayaram



Announcements

- **Showcase Info Form - due today - 11/12**

<https://forms.gle/CewUpMnmYq2BxupW6>

- **Optional React Zoom Session -
Friday 11/15 - Time TBD** (will be recorded)
- **Late Days** - 2 days maximum for Milestone 4 or HW3 (No need to send an email) subject to your attendance record.
- No late days for final project Milestone 5

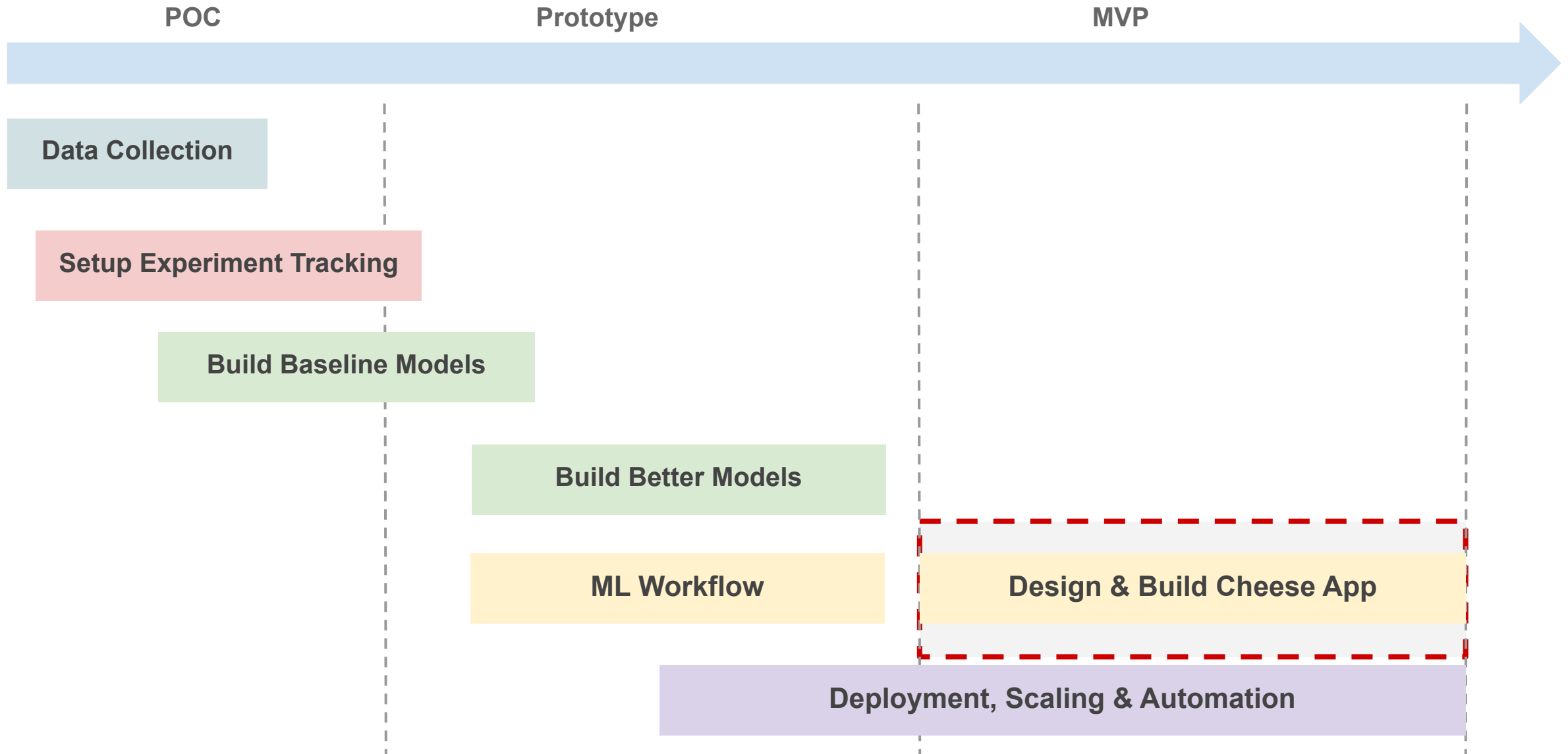
Outline

1. Recap
2. APIs
3. Frontend (Simple)
4. Frontend Frameworks
5. Frontend App (React)

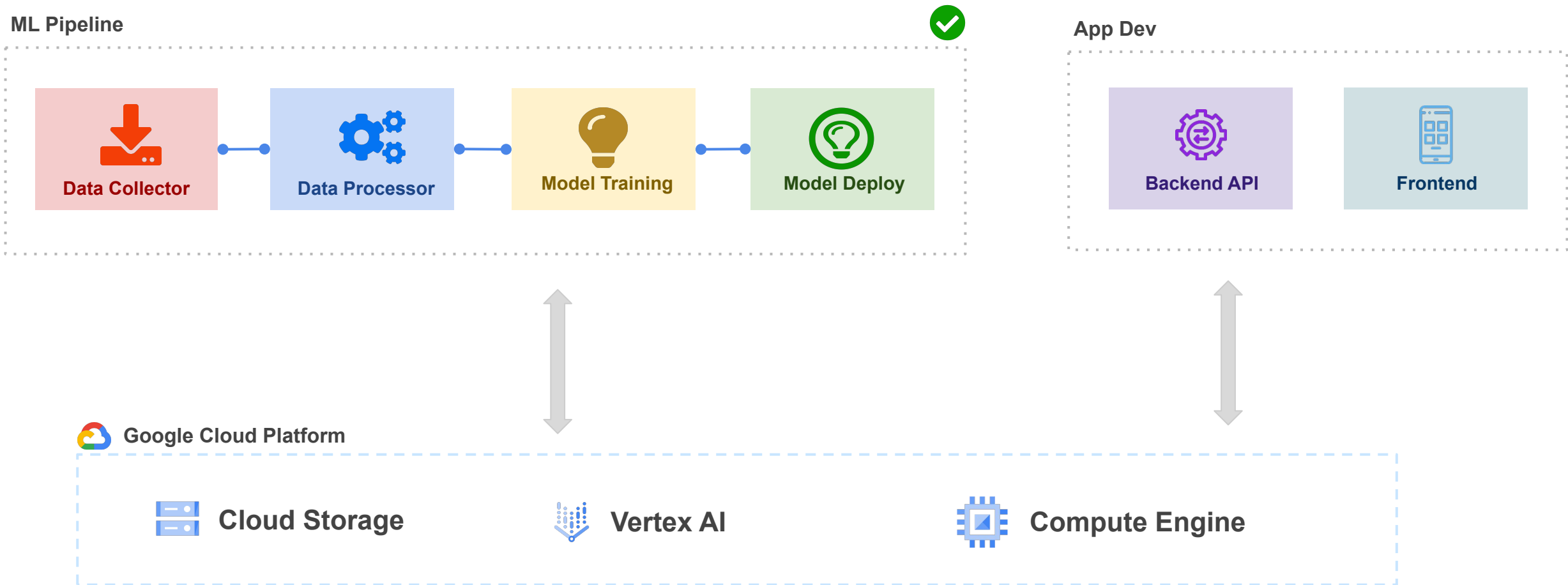
Outline

1. **Recap**
2. APIs
3. Frontend (Simple)
4. Frontend Frameworks
5. Frontend App (React)

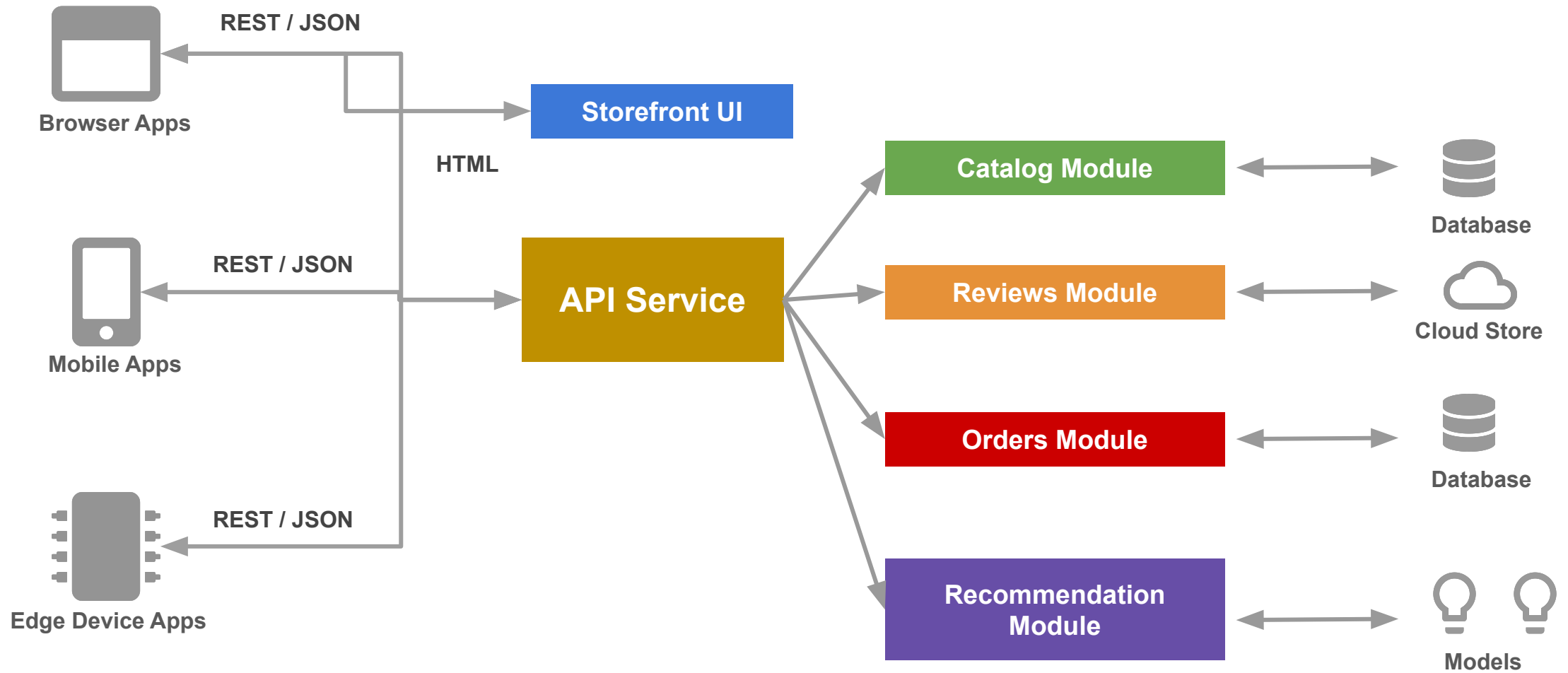
Recap: Cheese App Status



Recap: Cheese App Development



Recap: Microservice Architecture

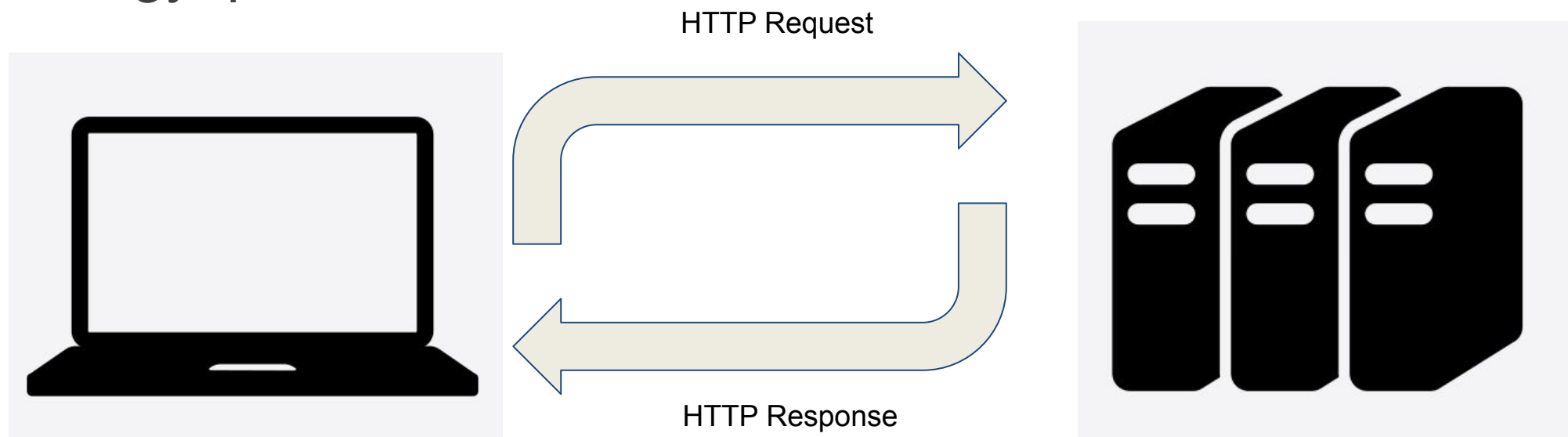


Outline

1. Recap
- 2. APIs**
3. App Frontend (Simple)
4. Frontend Frameworks
5. Frontend App (React)

Review: What is HTTP?

- HyperText Transfer Protocol: method for **transporting information** where **client** (such as a web browser) makes **request** and web **server** issues a **response**
 - content can be anything from text to images to video
- HTTPS: encryption for **secure** communication over network
- Analogy: post office



Review: What is a port?

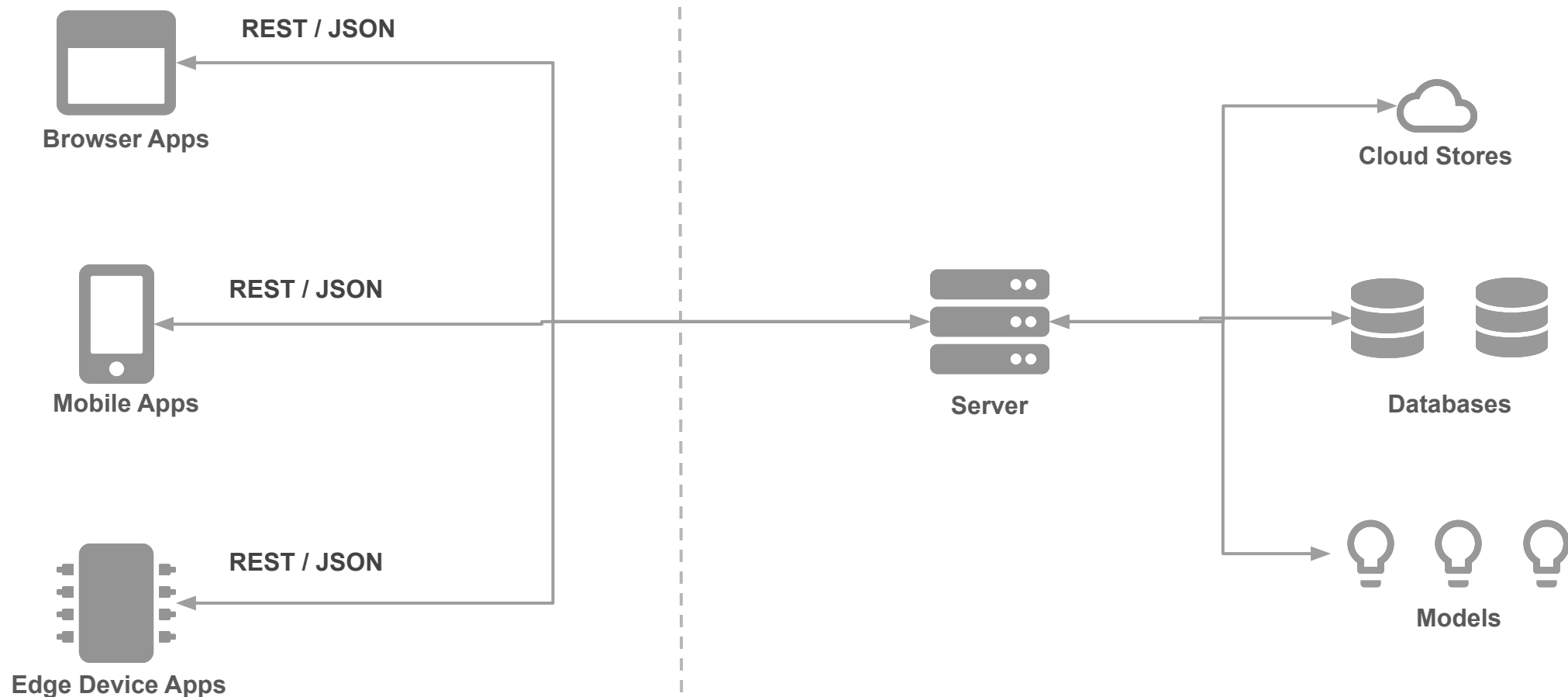
- **communication endpoint** where network connections start and end
- lets **computers differentiate** between different kinds of **data** (emails, webpages, etc.)
 - Port 22 = SSH
 - Port 25 = SMTP (email)
 - Port 80 = HTTP
 - Port 443 = HTTPS

What is an API

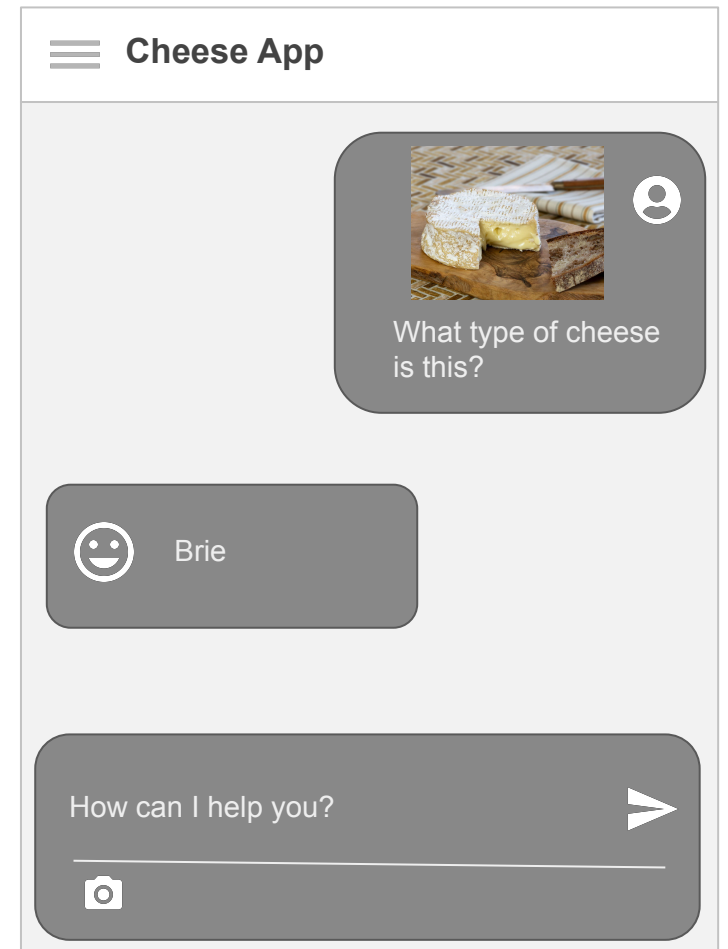
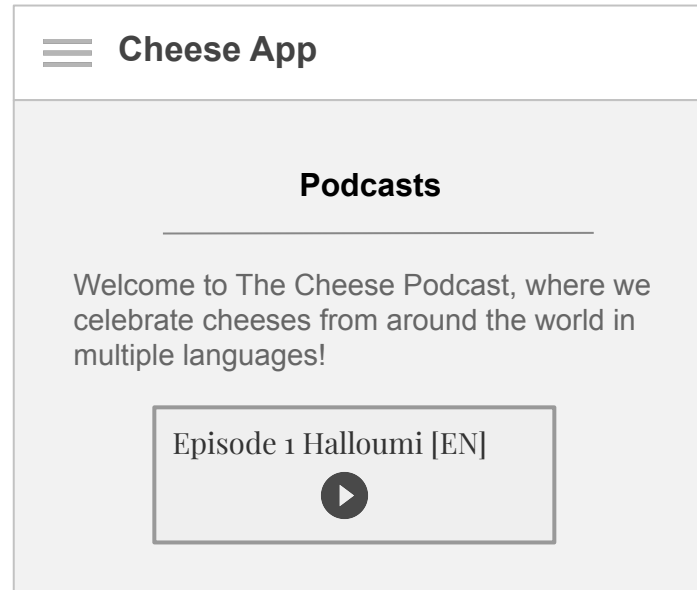
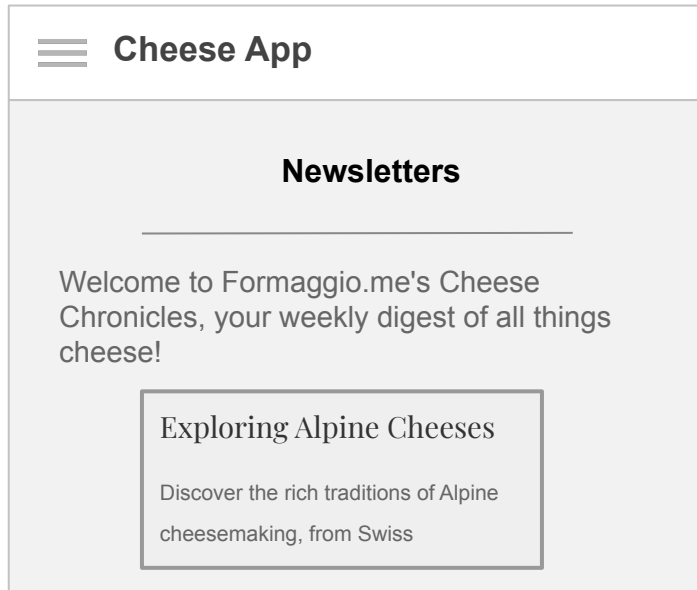
- API is **Application Programming Interface**
- **Web API** is an API that can be access using HTTP/S
- A **REST API** is a Web API that follows the HTTP method constraints - get, post, put, delete
- We will use **FastAPI** a Python framework to build REST APIs

APIs

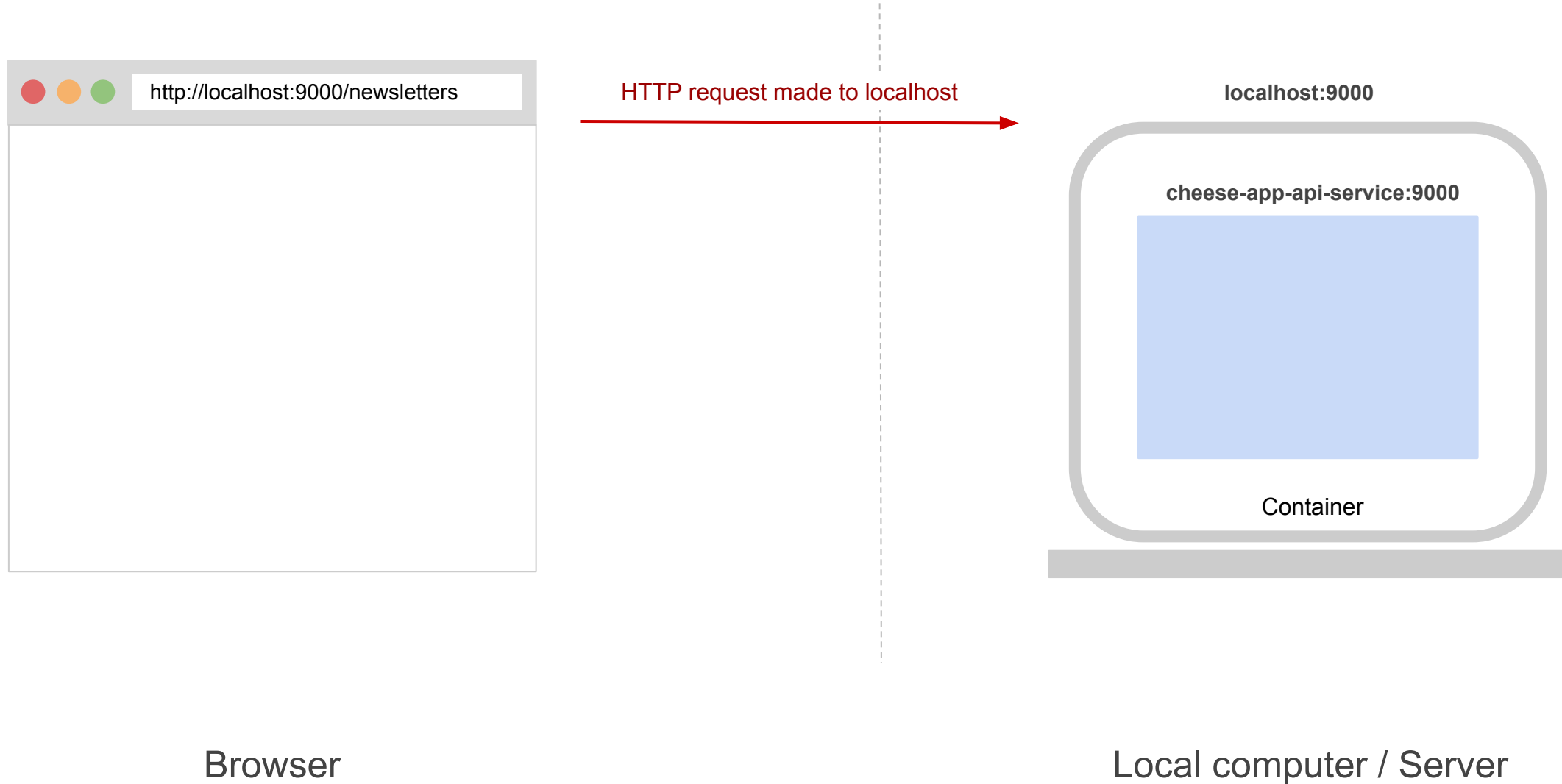
We will be using the term **API** to refer to REST API, which will be used to connect to various components



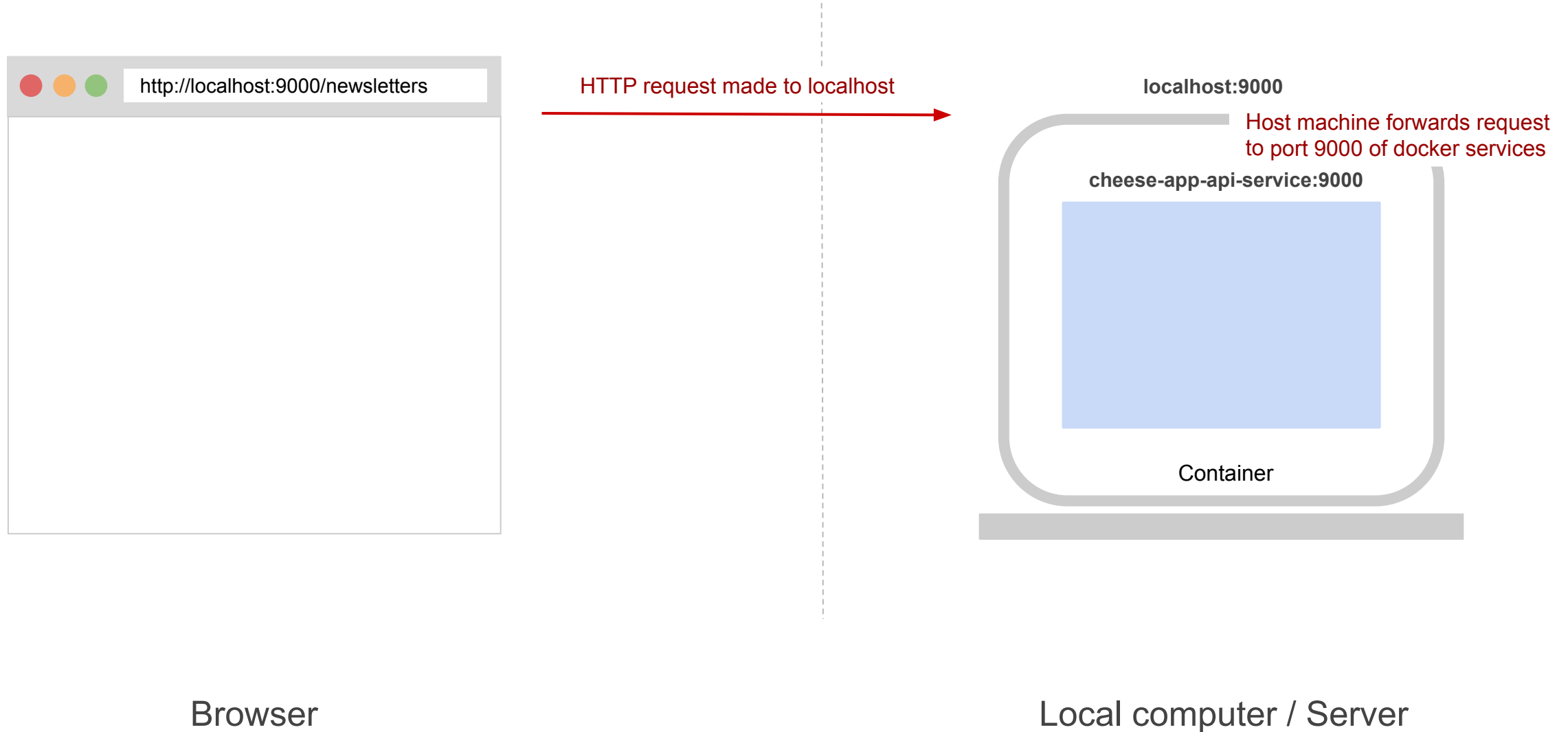
Review: Screenflow & Wireframes



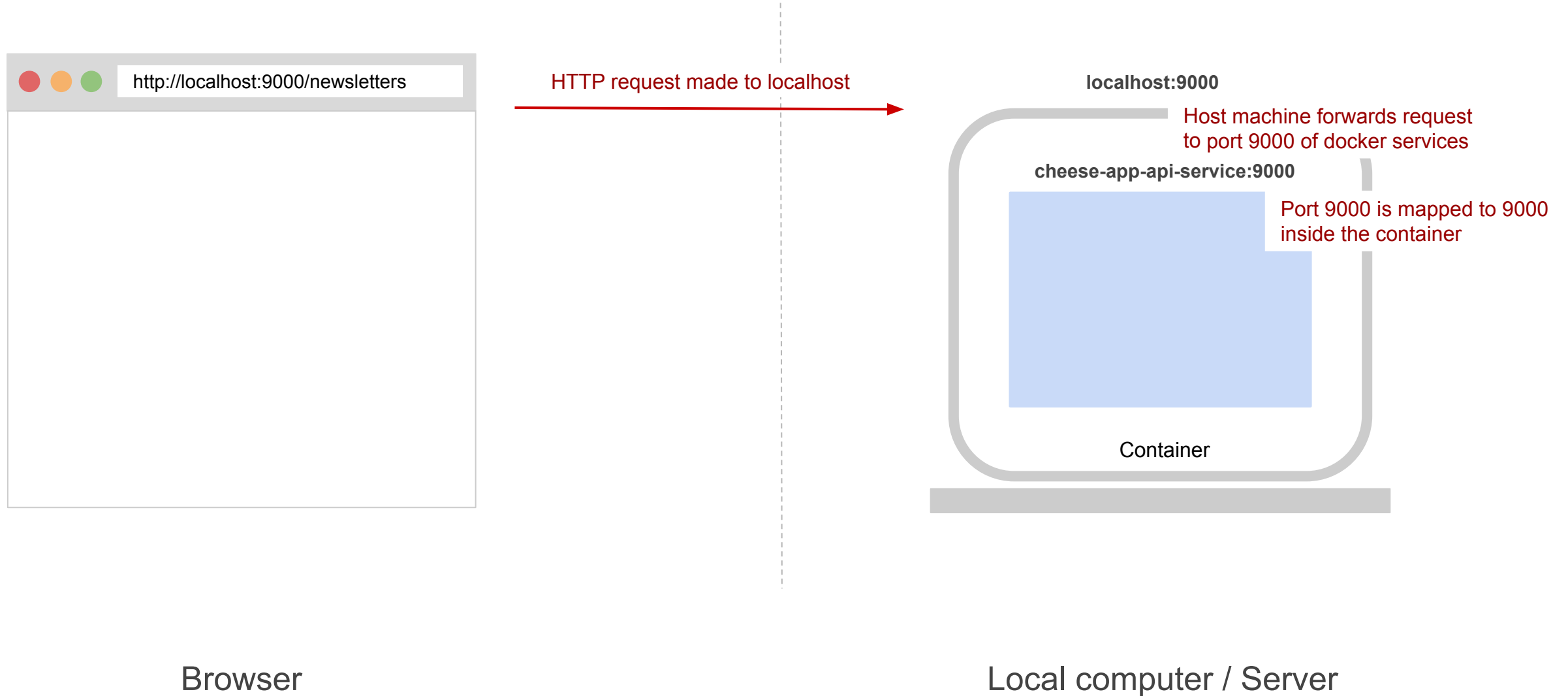
How does an API work



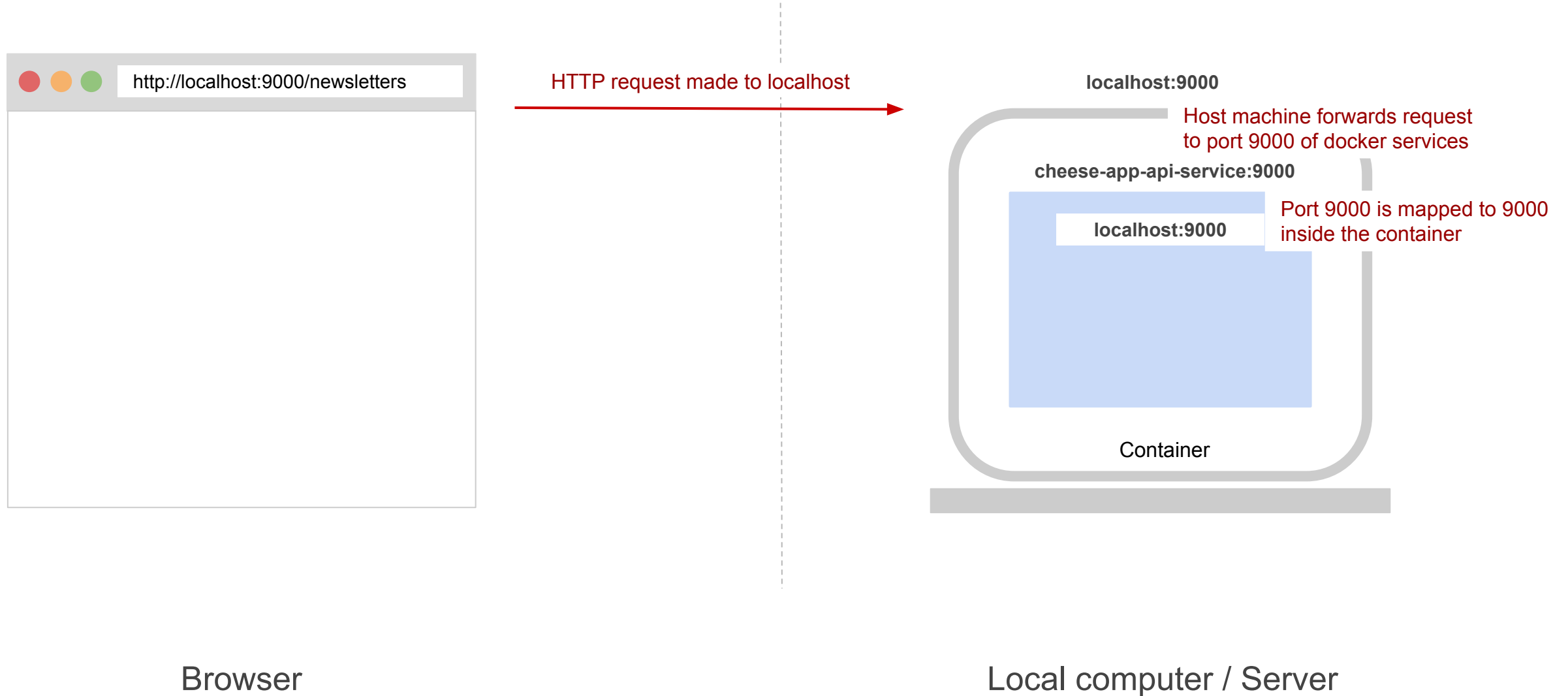
How does an API work



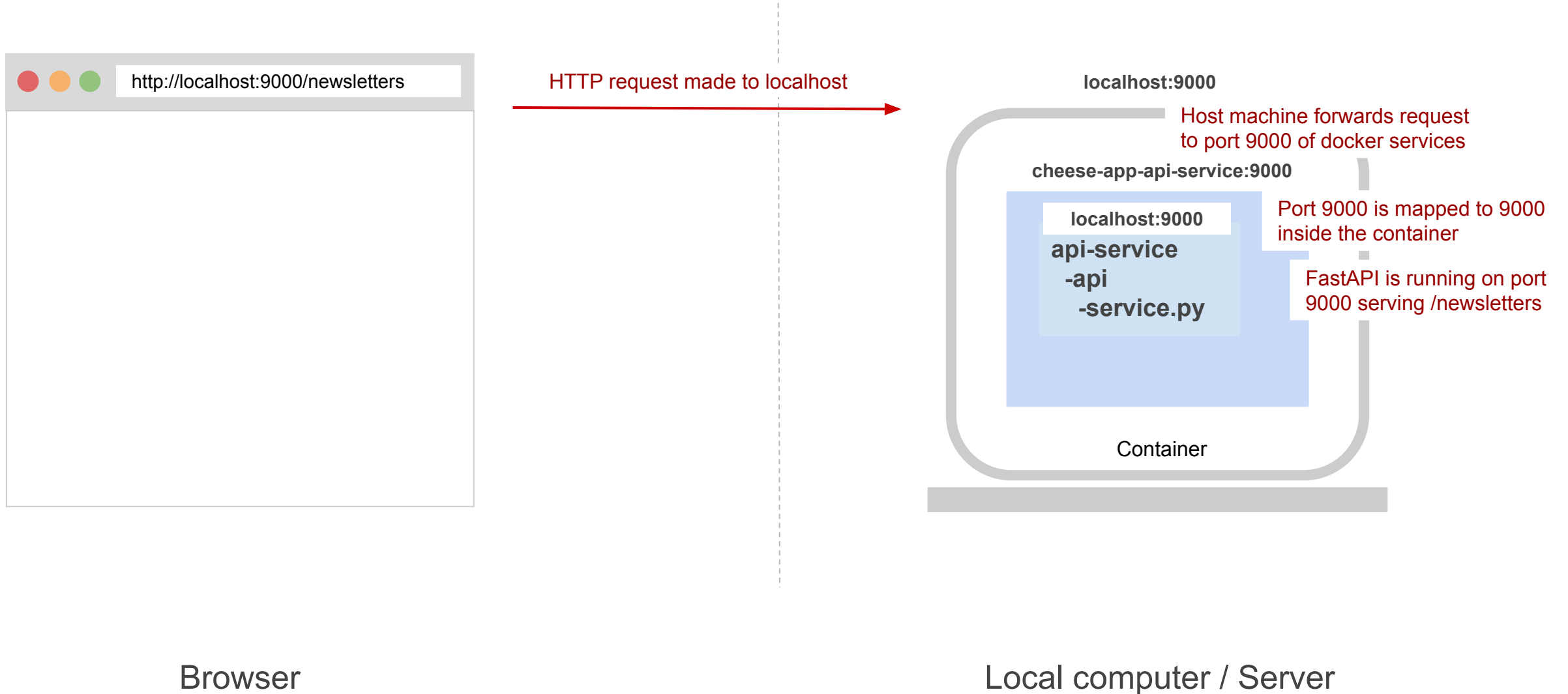
How does an API work



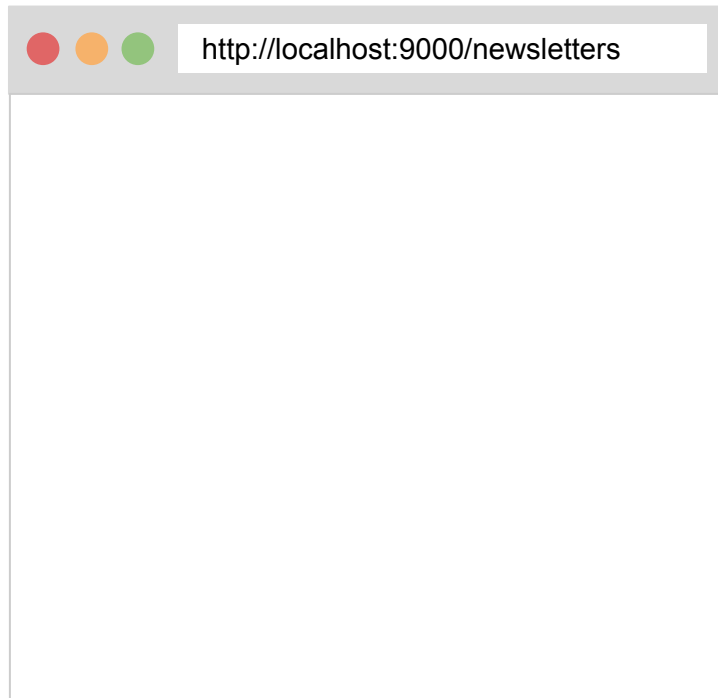
How does an API work



How does an API work

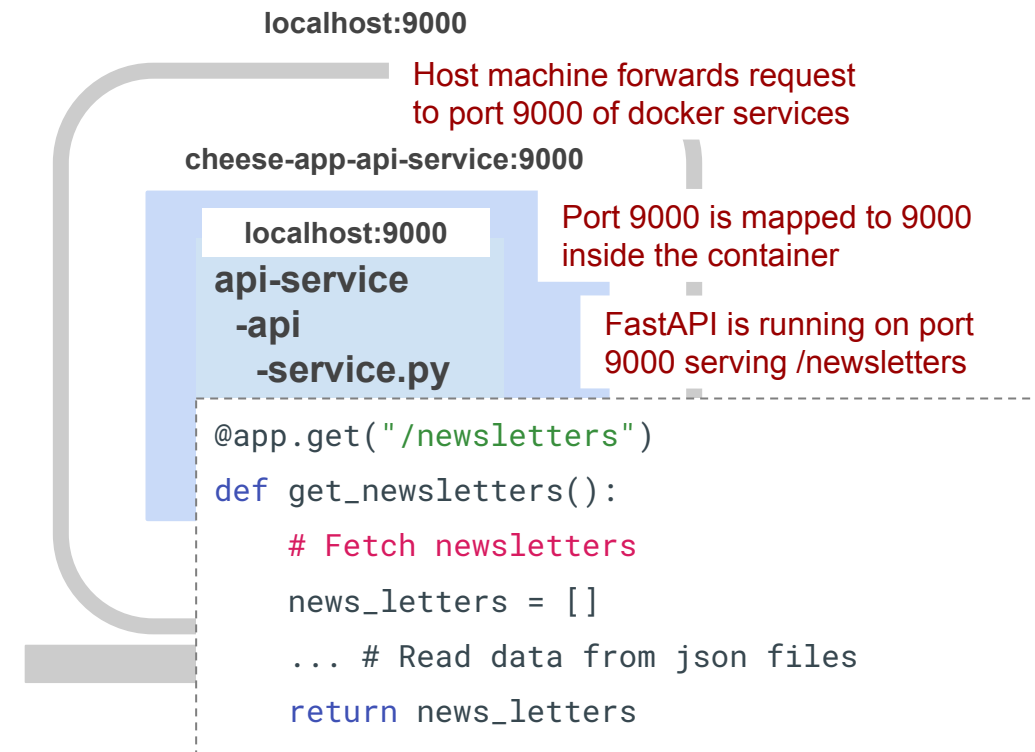


How does an API work



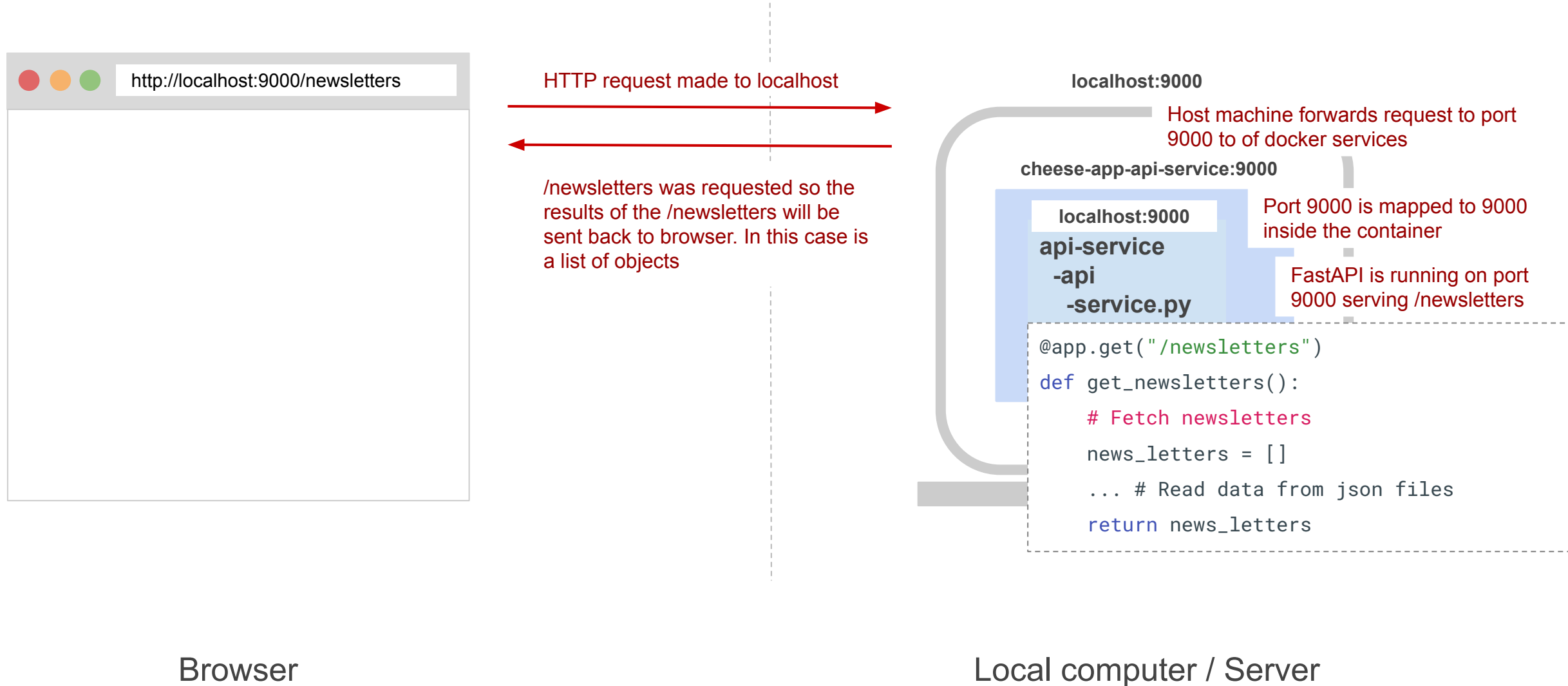
Browser

HTTP request made to localhost



Local computer / Server

How does an API work



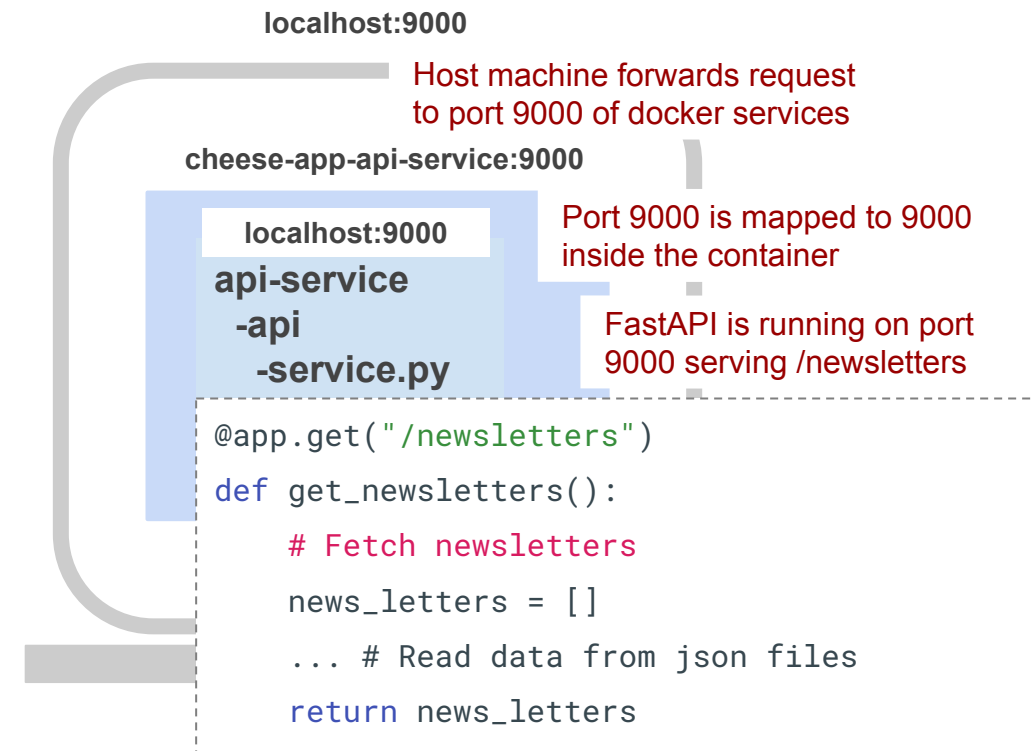
How does an API work



Browser

HTTP request made to localhost

`/newsletters` was requested so the results of the `/newsletters` will be sent back to browser. In this case is a list of objects



Local computer / Server

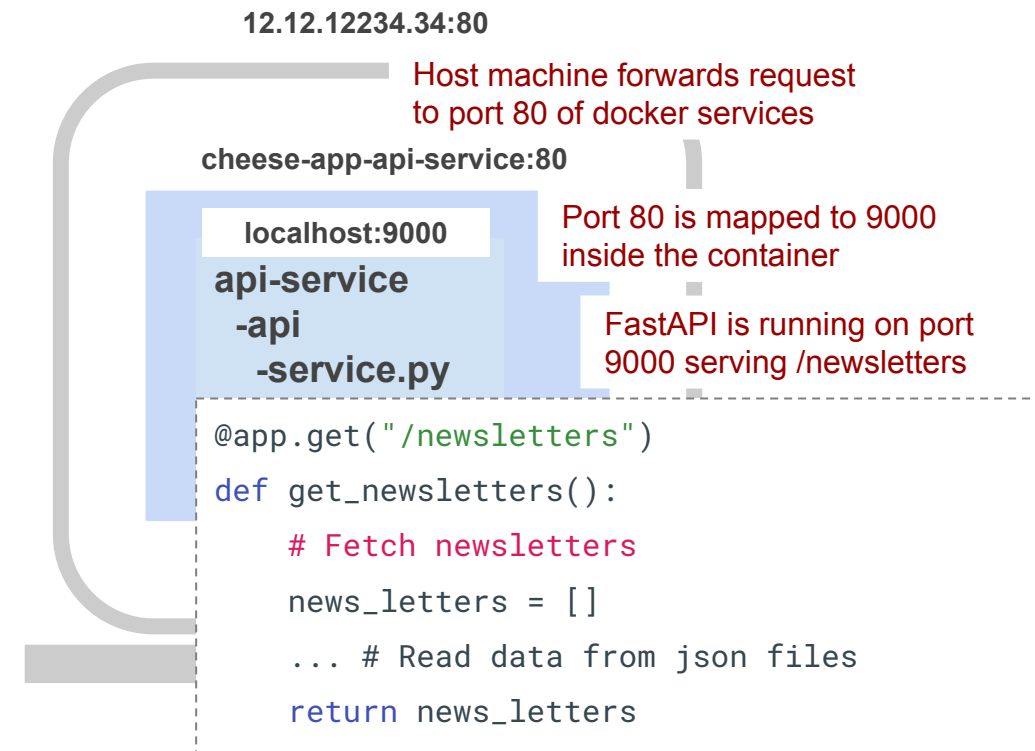
How does an API work (In Production)



Browser

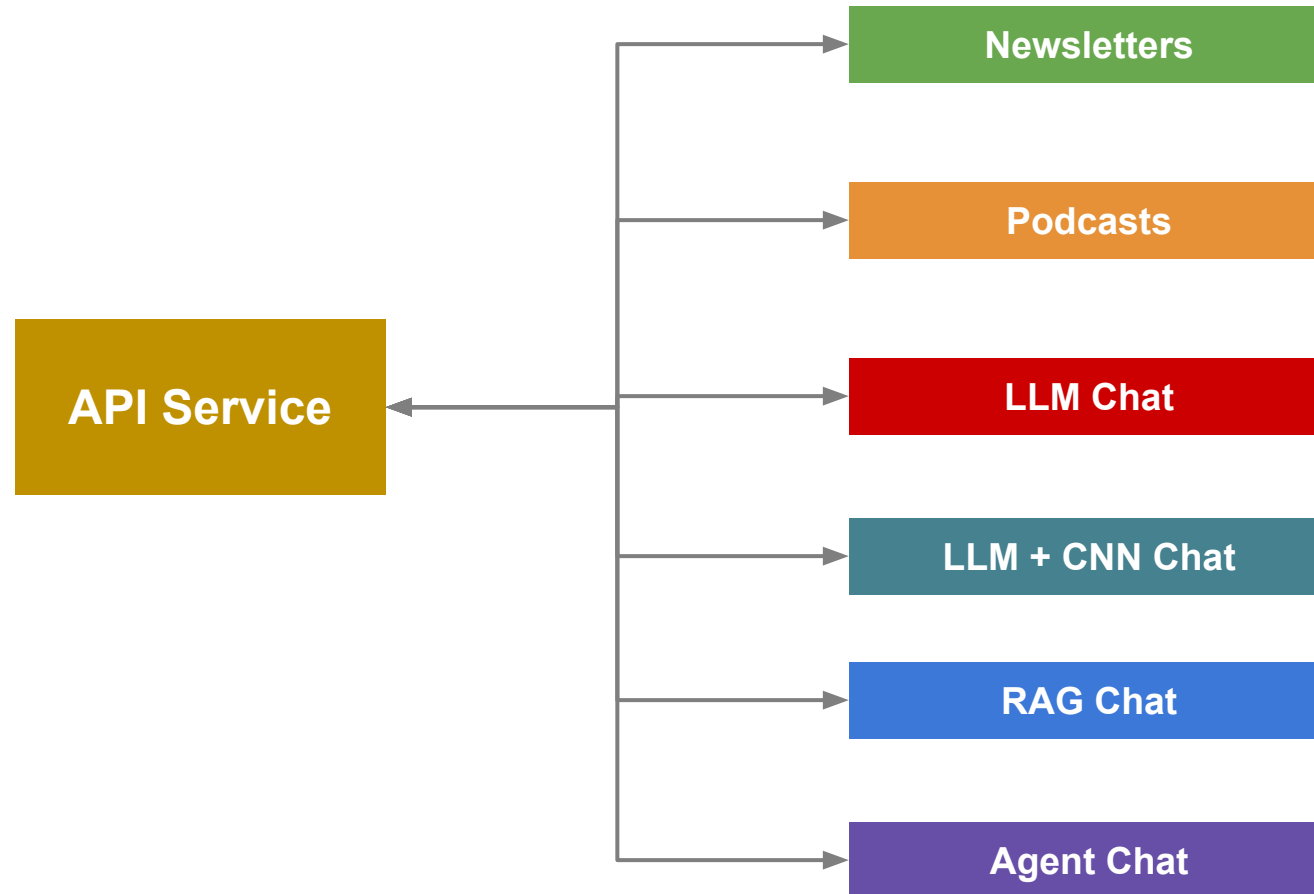
HTTP request made to server

/newsletters was requested so the results of the /newsletters will be sent back to browser. In this case is a list of objects



GCP Server

Tutorial: APIs



Tutorial: APIs

Steps to build Cheese App **APIs**:

- Ensure vector database is running.
- Expose data using an API.
- For detailed instructions, please refer to the following link
 - Cheese App APIs. (https://github.com/dlops-io/cheese-app-v2#setup-environments_)

Outline

1. Recap
2. APIs
3. **App Frontend (Simple)**
4. Frontend Frameworks
5. Frontend App (React)

HTML

- Is Hyper Text Markup Language (Remember Markdowns)
- Browsers use HTML to display web pages

CSS

- Cascading style sheets
- Used to format & style web pages

Javascript

- Programming language understood by browser

App Frontend

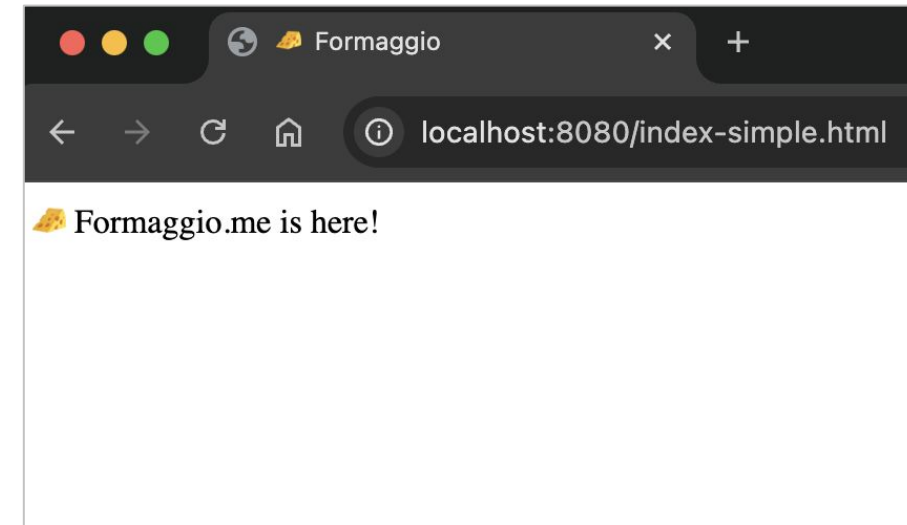
```
<!DOCTYPE html>
<html>
<head>
  <title>🧀 Formaggio</title>
  <style>body{background-color: #efefef;}</style>
</head>
<body>
  🧀 Formaggio.me is here!
</body>
<script>
  var input_file =
document.getElementById("input_file");
</script>
</html>
```

Browser Title

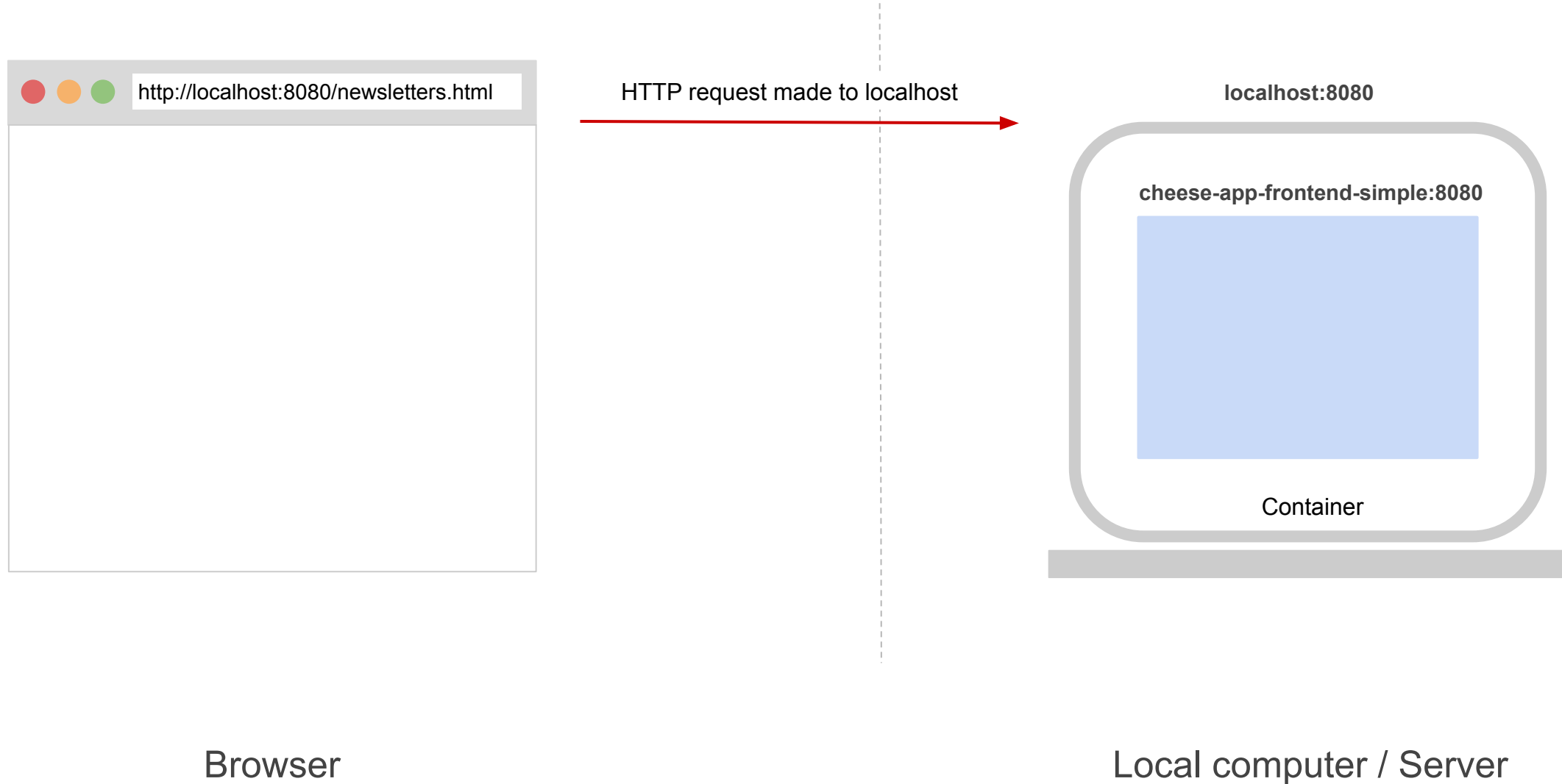
Page Style

Web page details

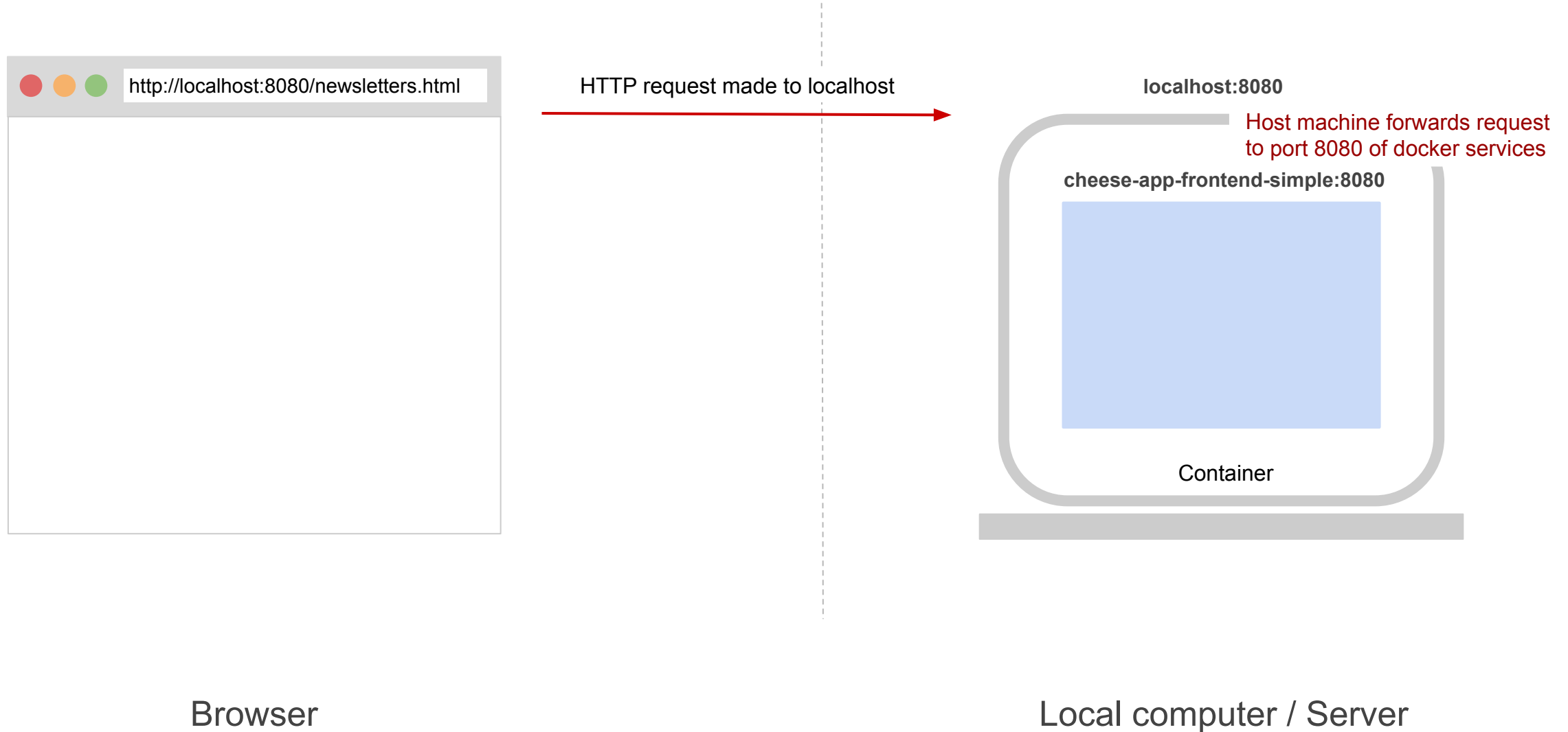
Web page scripts (Javascript)



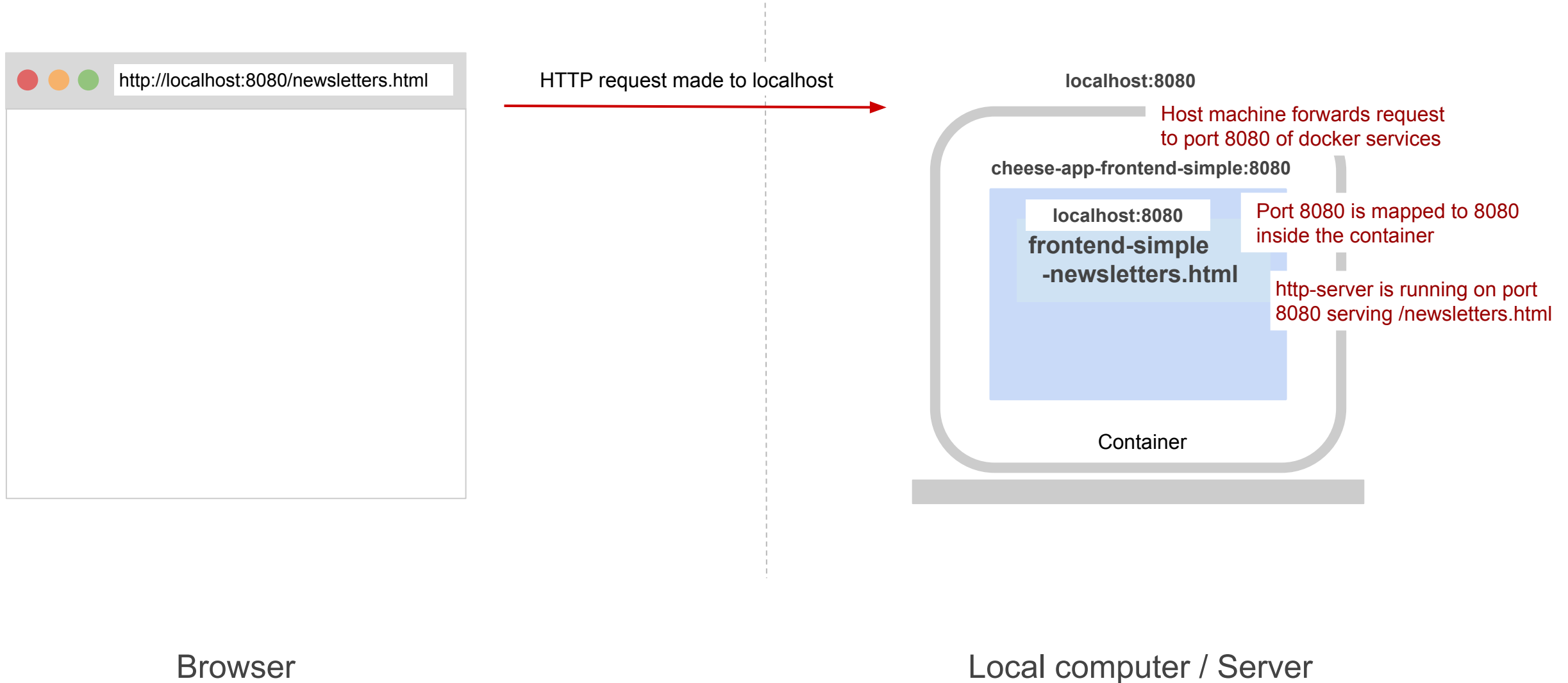
How does the App work



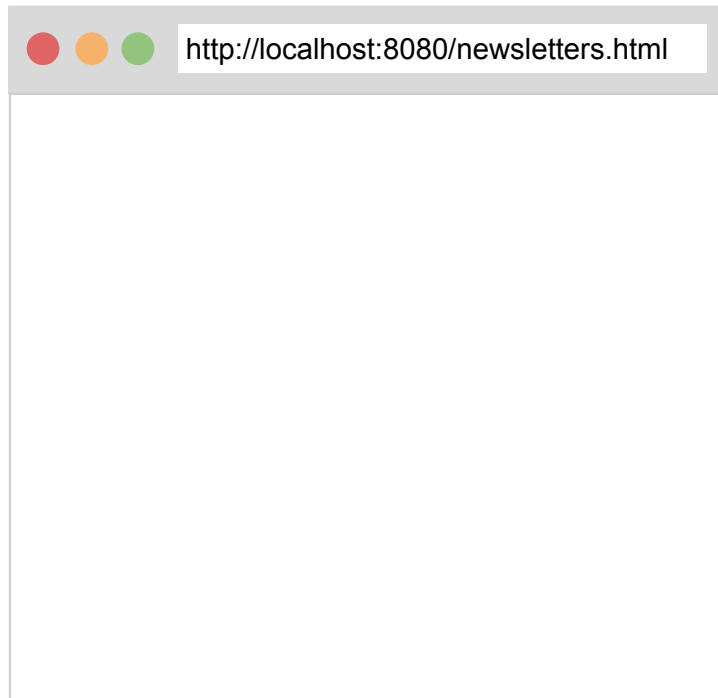
How does the App work



How does the App work

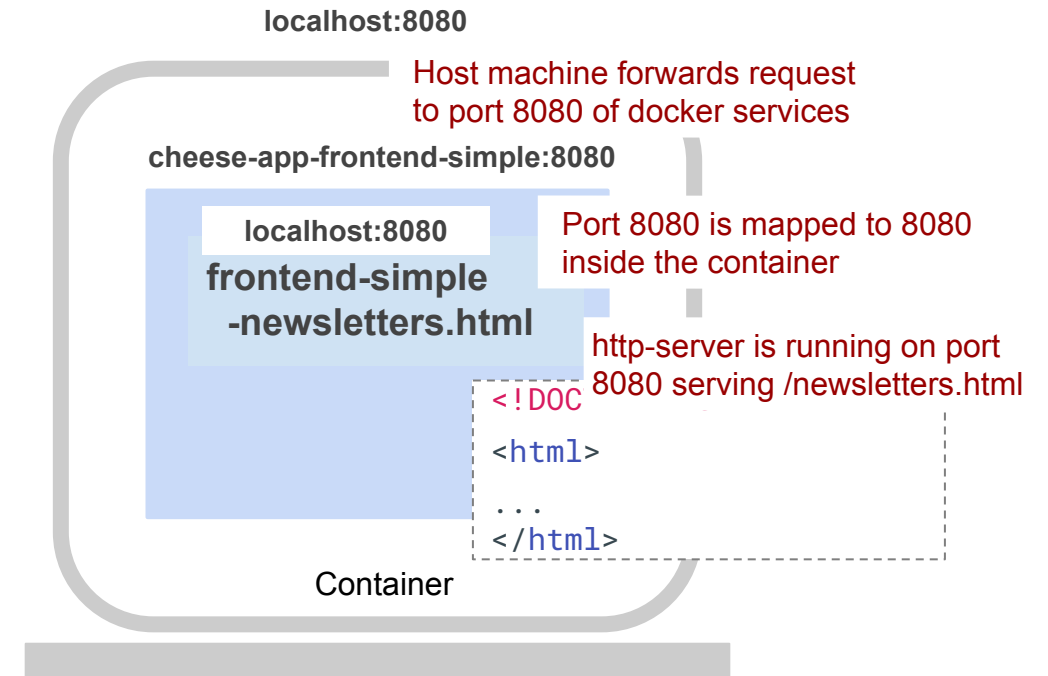


How does the App work



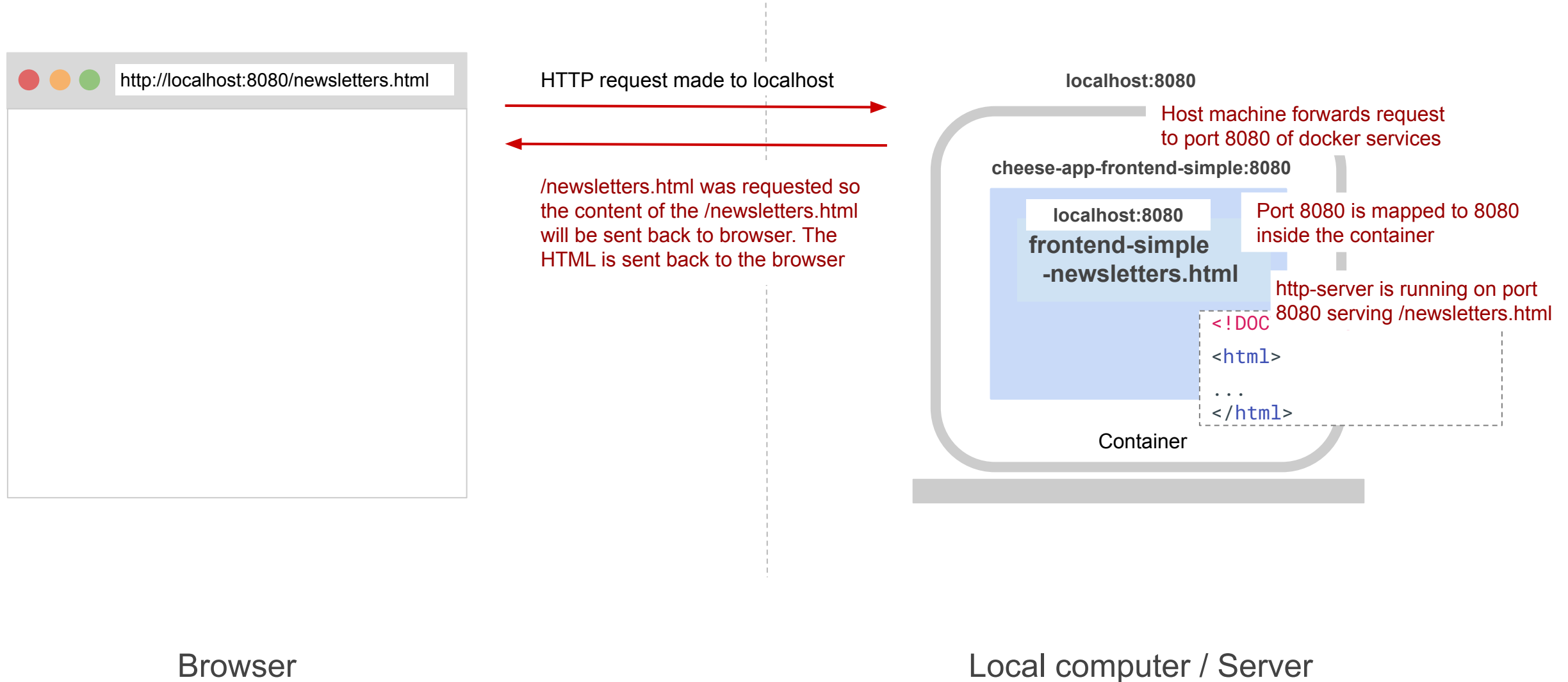
Browser

HTTP request made to localhost

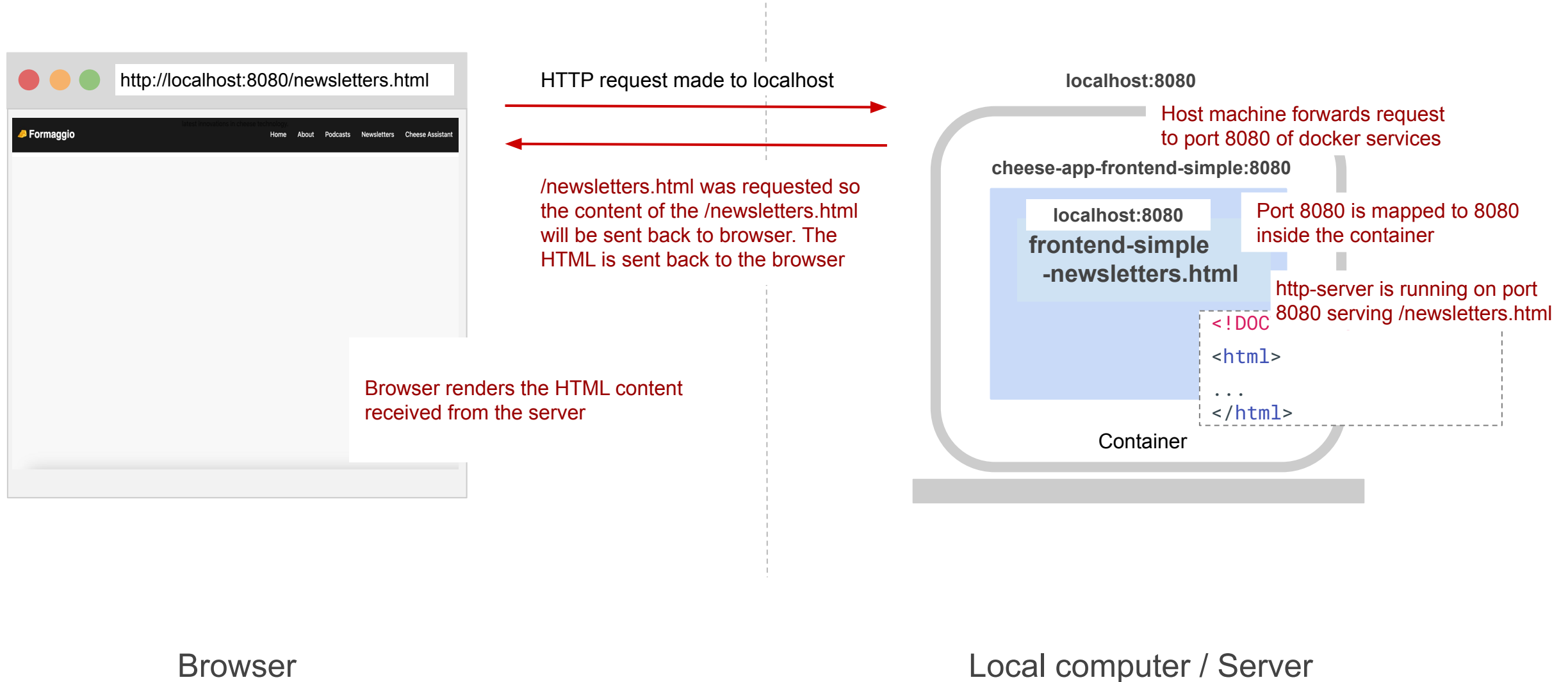


Local computer / Server

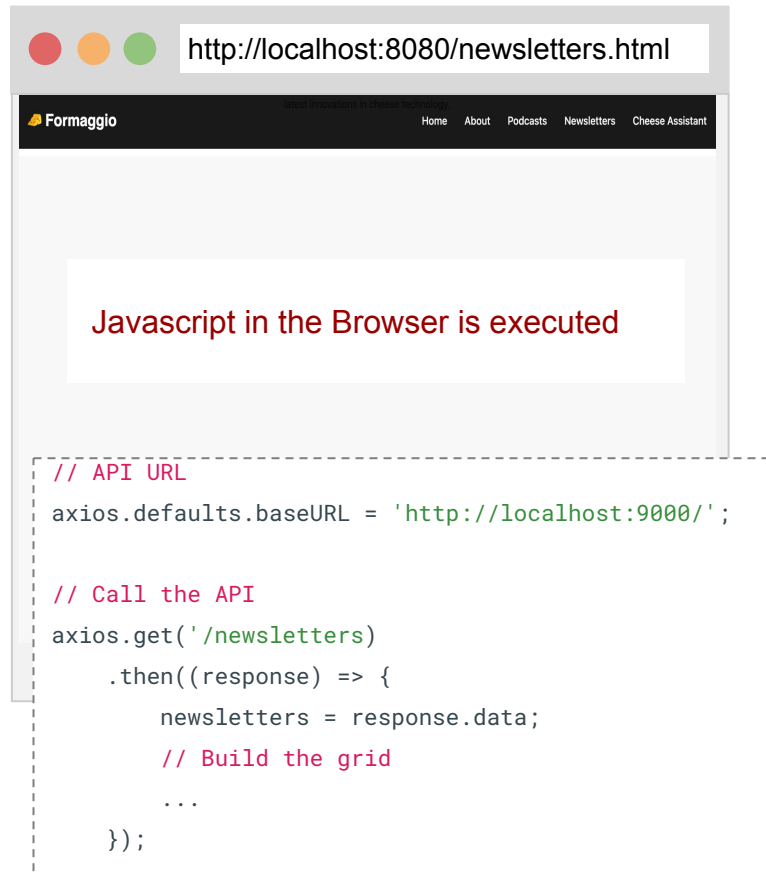
How does the App work



How does the App work

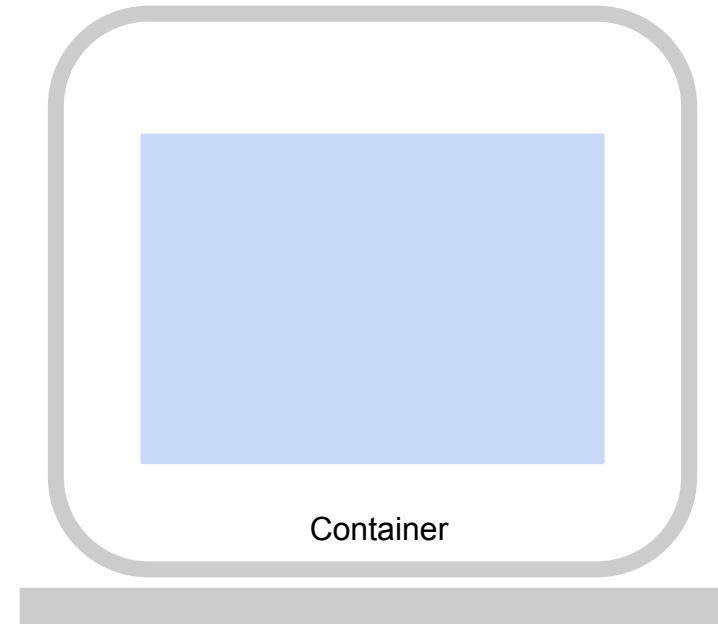


How does the App work



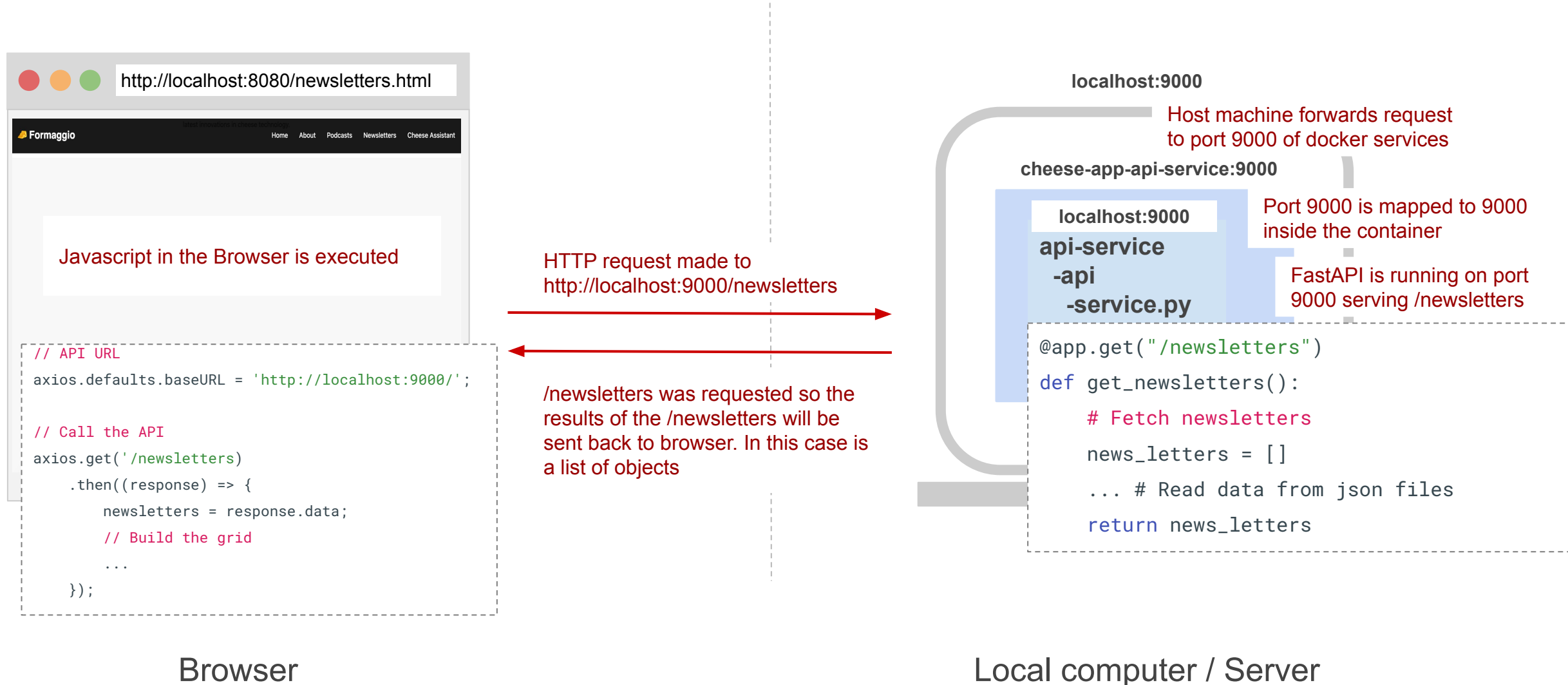
Browser

HTTP request made to
`http://localhost:9000/newsletters`

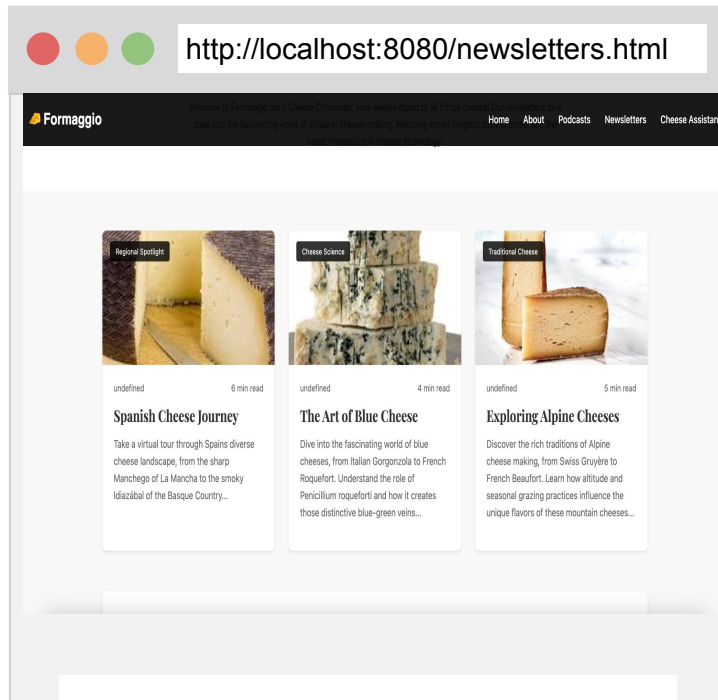


Local computer / Server

How does the App work



How does the App work

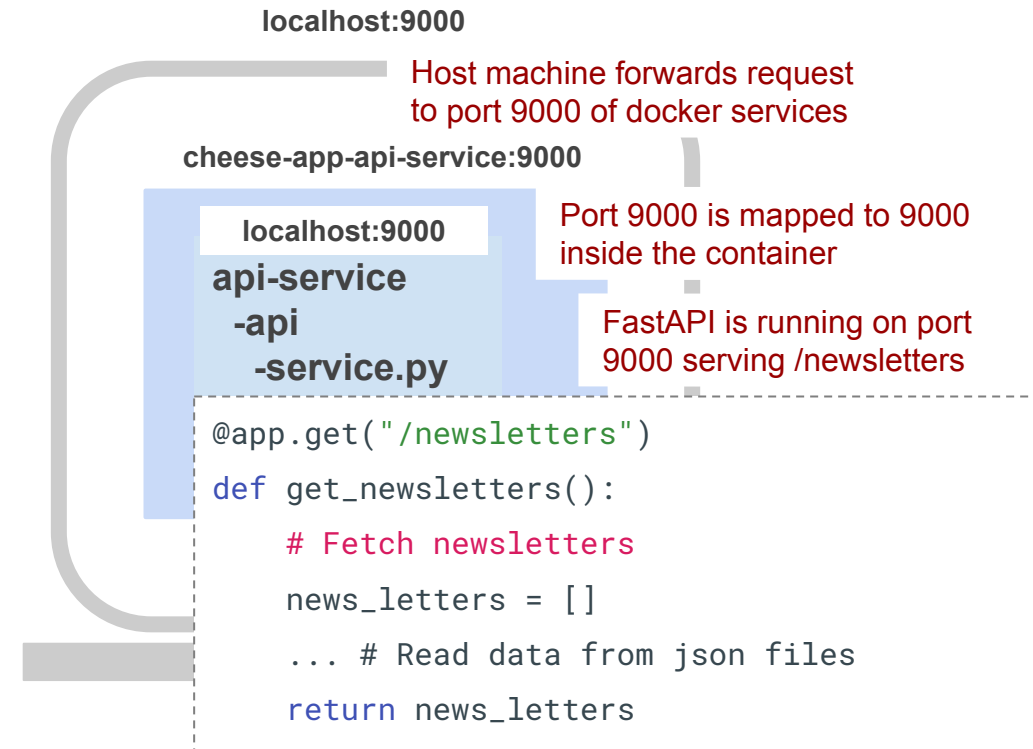


Javascript displays the newsletters data in the html page.

Browser

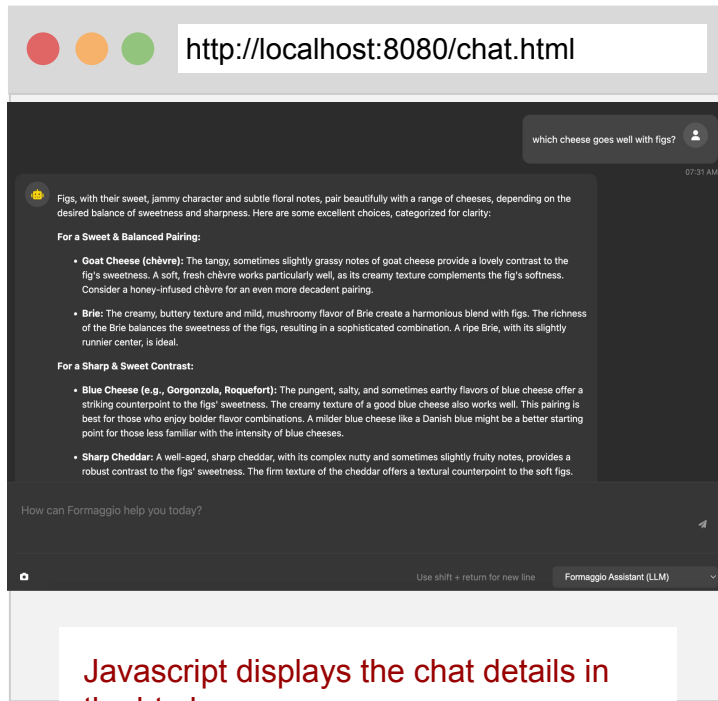
HTTP request made to
`http://localhost:9000/newsletters`

`/newsletters` was requested so the results of the `/newsletters` will be sent back to browser. In this case is a list of objects



Local computer / Server

How does the App work (Chat)

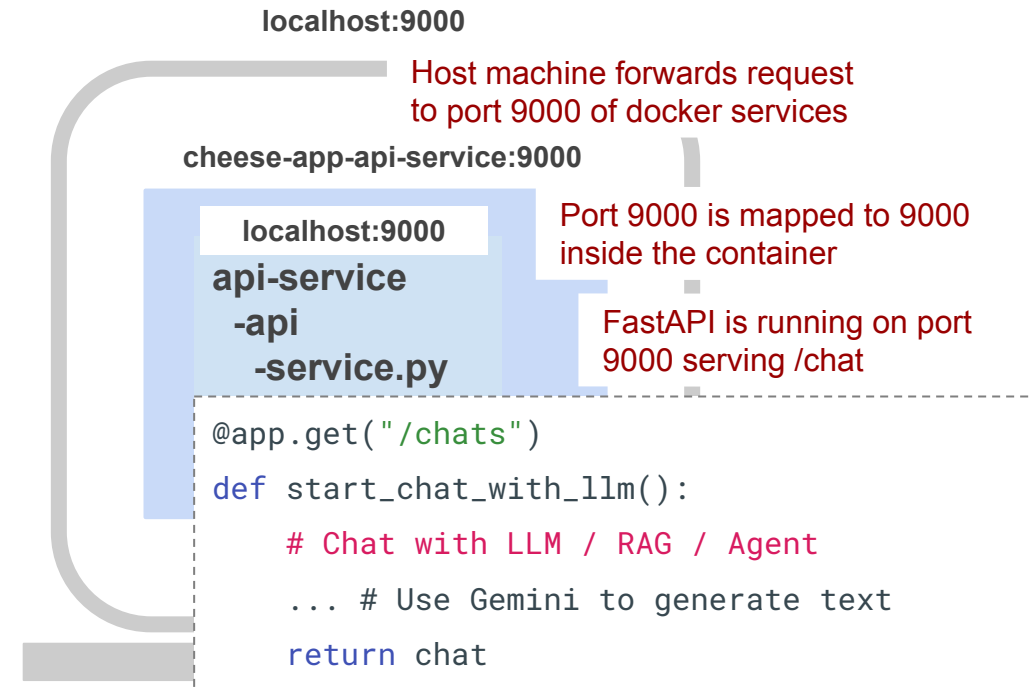


Javascript displays the chat details in the html page.

Browser

HTTP request made to
http://localhost:9000/chat

/chat was requested so the results
of the /chat will be sent back to
browser. In this case it is the chat
details



Local computer / Server

Tutorial: Frontend Simple

Steps to run Cheese App **Frontend**:

- <https://github.com/dlops-io/cheese-app-v2#frontend-app-simple>

Outline

1. Recap
2. APIs
3. App Frontend (Simple)
4. **Frontend Frameworks**
5. Frontend App (React)

Frontend

When we build our frontend we need a page for each component:

- index.html
- newsletters.html
- podcasts.html
- chat.html

Frontend

When we build our frontend we had a page for each component:

- index.html
- newsletters.html
- podcasts.html
- chat.html

Problems:

- Each of these had its own HTML, Javascript, CSS
- How do we share/reuse code across pages?
- Each page is loaded separately in browser (Slow)

Frontend

Problems:

- Each of these had its own HTML, Javascript, CSS
- How do we share/reuse code across pages
- Each page is loaded separately in browser (Slow)

Solution:

- Create a single page app that manages HTML, Javascript, CSS as components
- Use frontend App **Frameworks**

Frontend Frameworks

The common frontend app frameworks are:

- Angular (Google)
- **React (Facebook)**
- Vue
- Svelte

React

- Everything is a **Component**
- Uses **JSX** instead of Javascript
- JSX is an extension to JavaScript
- JSX is like a template language, but it comes with the full power of JavaScript

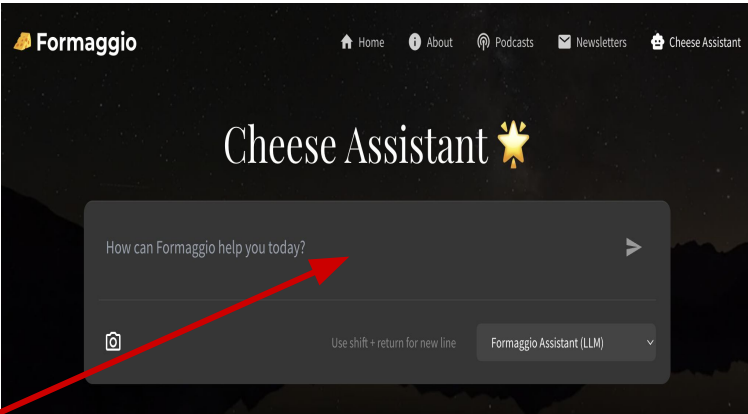
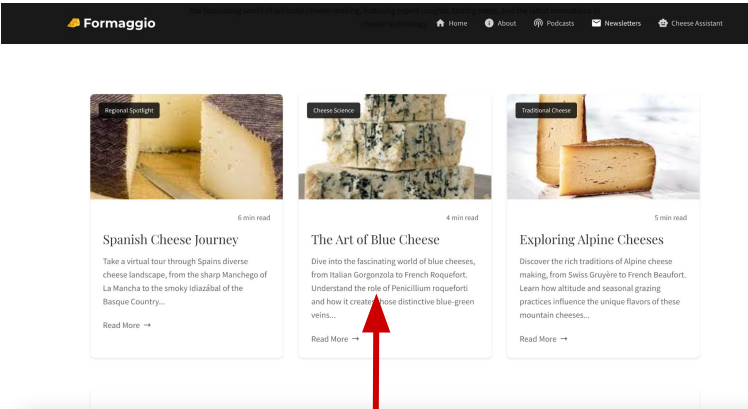
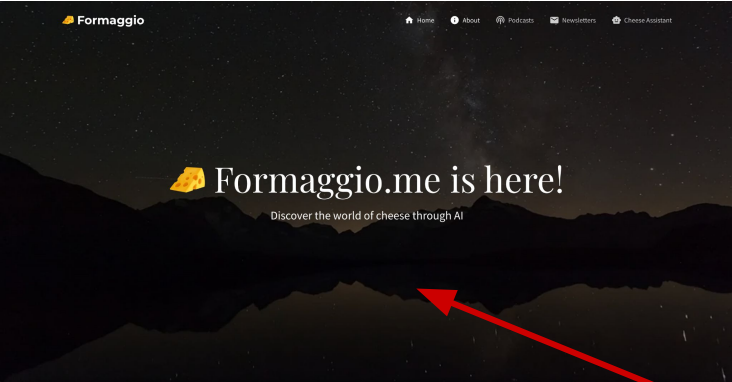
Header

Content

Footer

React App

Header defined only once



Content block switched for each page

Tutorial: Frontend React

Steps to run Cheese App **React Frontend**:

- <https://github.com/dlops-io/cheese-app-v2#frontend-app-react>

THANK YOU