Overview

The Model-Template-View pattern

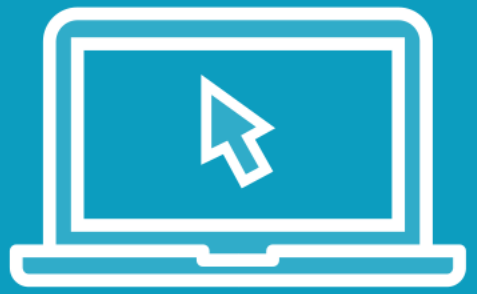
Add Jinja templates

Add a data model

- Text file (JSON), no actual DB

How do the MTV components interact?

Demo



Jinja templates

- Displaying data to the user
- Generating HTML
- Calling templates from view
- Passing data from view to template
- Jinja variables

Jinja Templates

Templates

- Components that *display* data to the user

Jinja templates generate text files

- We use it to create HTML

Templates are text files

- With placeholders for variables
- And logic: if, for, filters, etc.

```
<html><head><title>{{name}}'s Page</title></head>
<body>
    <h1>Hi I'm {{name}}!</h1>
        I'm {{age}} years old.
</body></html>
```

A Jinja Template Example

{{ var }} looks up var in the *template context*

Other text is copied to HTML output

This creates *dynamic* HTML page

```
from flask import render_template


def welcome():
    return render_template("welcome.html",
        name="Bob", age=35)
```

Calling a Template from a View

render_template: first argument is name of template file

Other arguments are data passed to the *template context*

Templates should be in folder /templates

Don't forget the return keyword

Demo

Adding a model layer

- Represents our data
- Text file instead of database

Model-Template-View

Model

Data

Usually from DB

Flask: leaves it to us

Template

Presentation

Generates HTML

In Flask: Jinja2

View

Behaviour

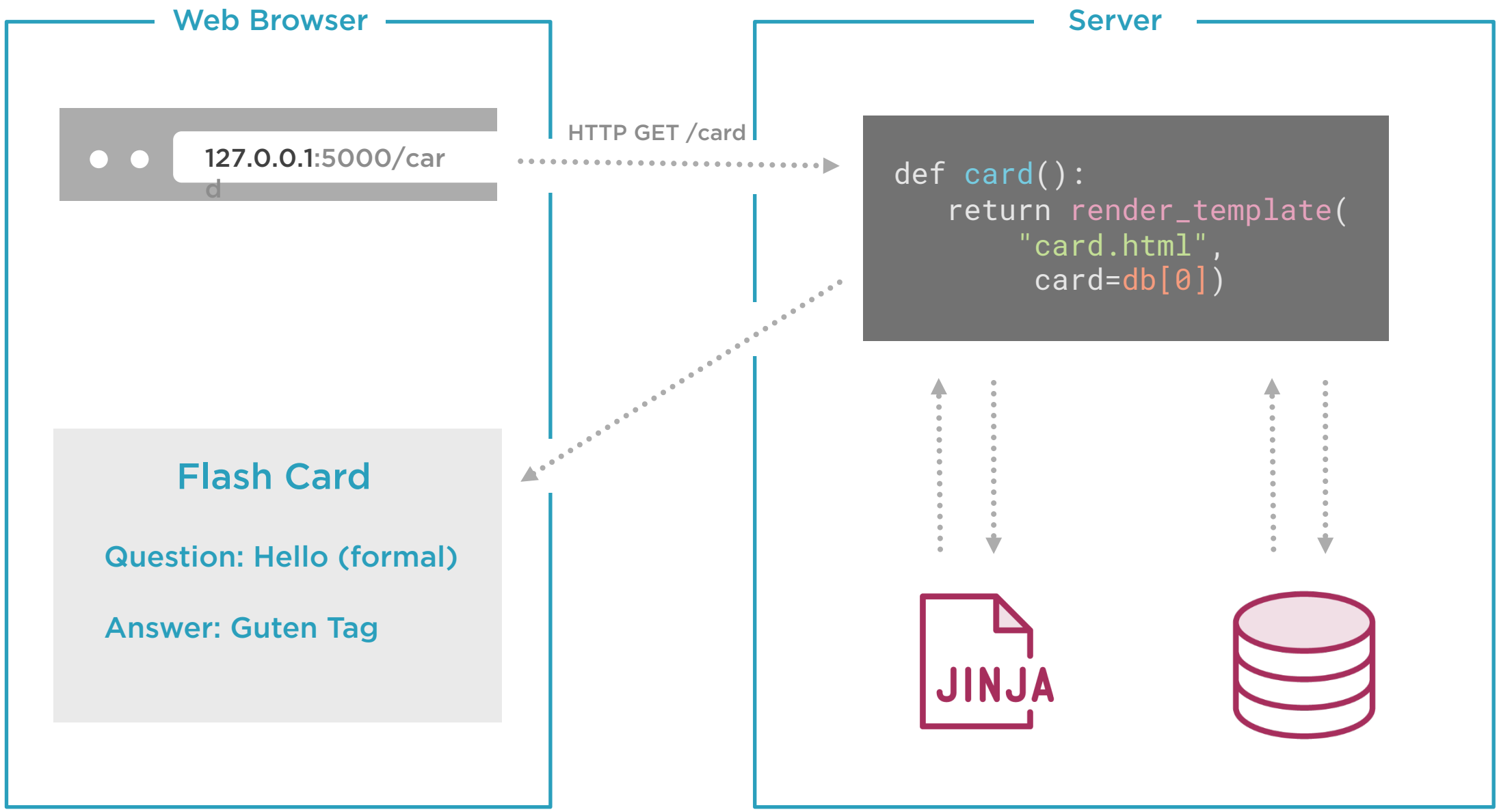
Python function

Mapped to URL

Model

View

Controller


Web Browser
127.0.0.1:5000/card
HTTP GET /card
Server
def card():
return render_template(
"card.html",
card=db[0])
Flash Card
Question: Hello (formal)
Answer: Guten Tag
JINJA

Summary

The Model-Template-View pattern

Jinja templates

- Jinja variables
- Calling templates from views
- Sending data from view to template

Add a data model

- Text file (JSON), no actual DB