

This is CS50

Week 9

Yuliia Zhukovets

Preceptor

yuliia@cs50.harvard.edu

Agenda

- Routes
- Request Methods
- Forms
- Updating a Model
- Template Rendering
- Problem Set Advice

Birthdays

×

+

←

→

↻

127.0.0.1:5000

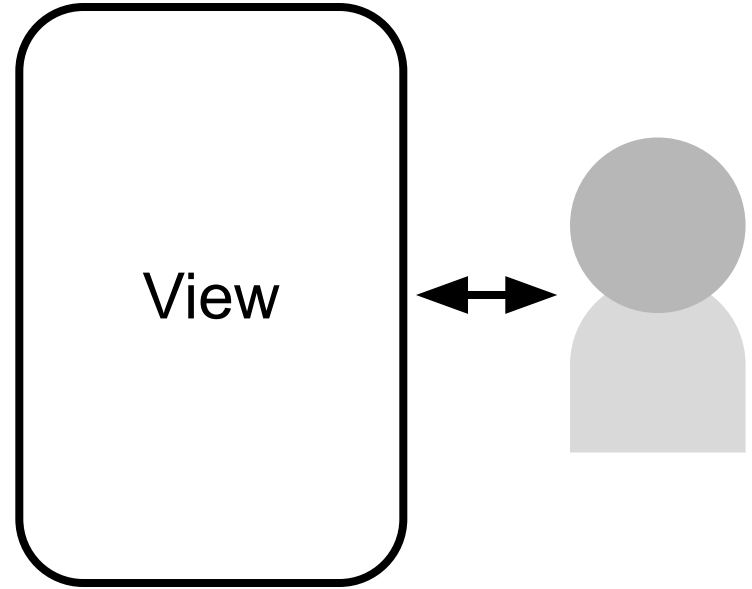
Guest

Birthdays

Add a Birthday

Add Birthday

All Birthdays



Birthdays

127.0.0.1:5000

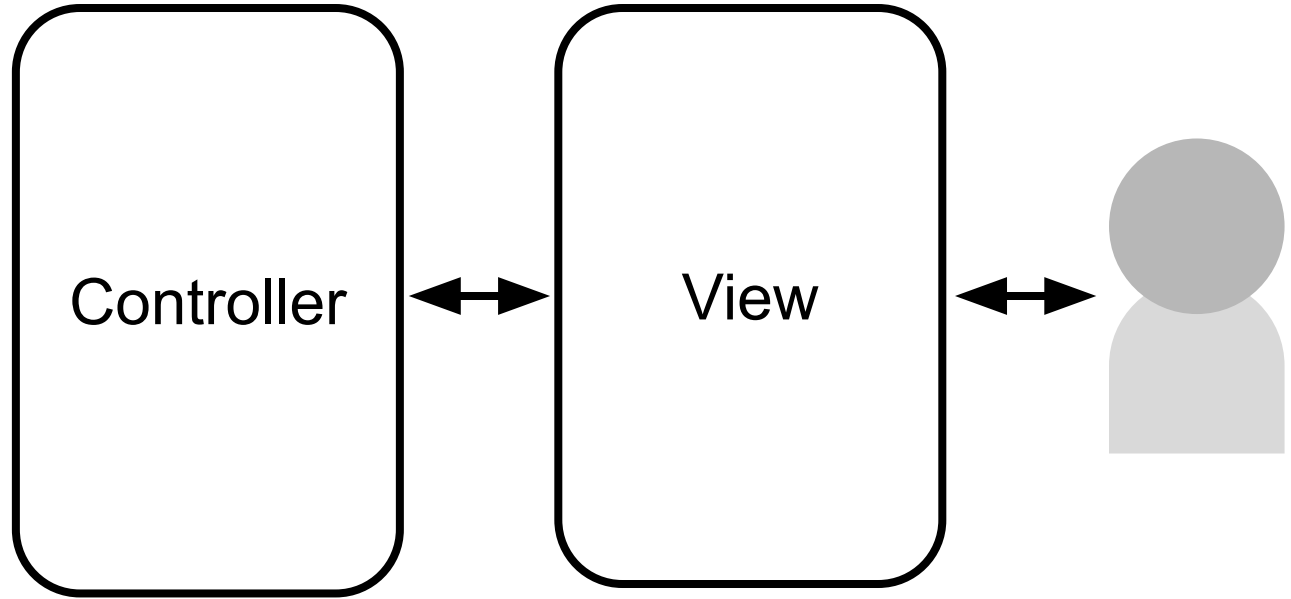
Guest

Birthdays

Add a Birthday

All Birthdays

Name	Birthday
Harry	7/31
Ron	3/1
Hermione	9/19



Birthdays

127.0.0.1:5000

Guest

Birthdays

Add a Birthday

Name

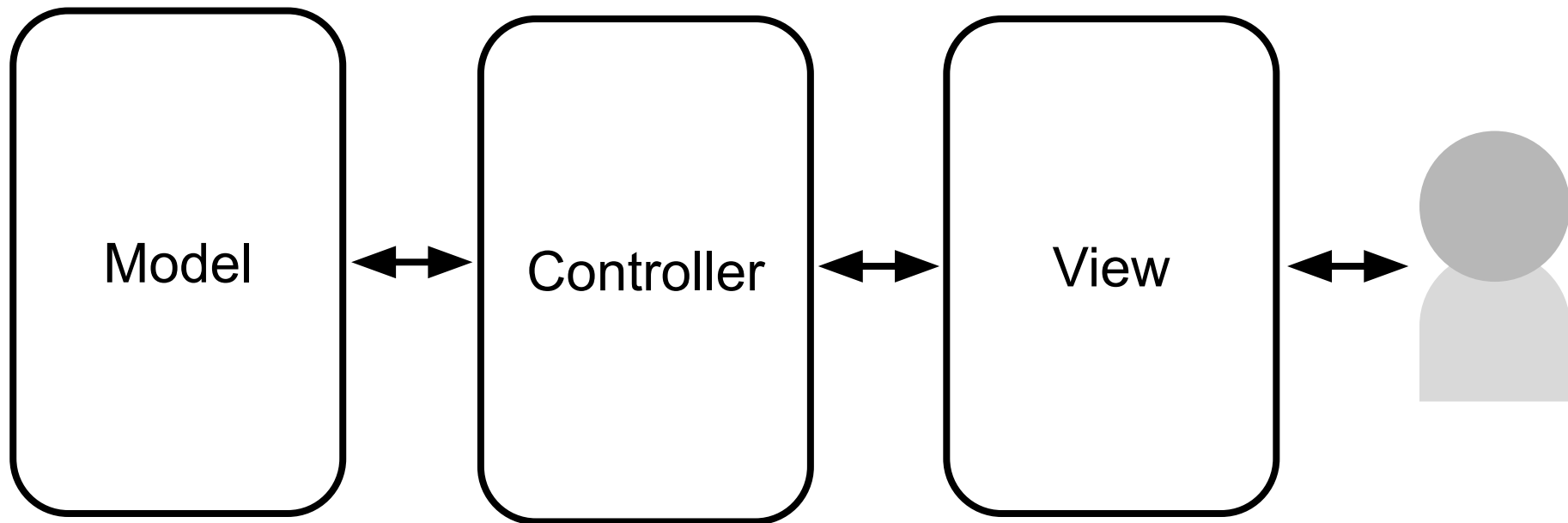
Month

Day

Add Birthday

All Birthdays

Name	Birthday
Harry	7/31
Ron	3/1
Hermione	9/19



Birthdays

127.0.0.1:5000

Guest

Birthdays

Add a Birthday

Add Birthday

All Birthdays

Name	Birthday
Harry	7/31
Ron	3/1
Hermione	9/19

birthdays/
 static/
 styles.css
 templates/
 index.html
app.py
birthdays.db

Routes

<https://birthdays.net/>

<https://birthdays.net/add>

```
@app.route("/")  
def index():  
    # Return index page ("homepage")
```

```
@app.route("/")  
def index():  
    return render_template("index.html")
```



```
flask run
```

Request Methods

```
@app.route("/")  
def index():  
    return render_template("index.html")
```

```
@app.route("/", methods=["GET", "POST"])
def index():
    return render_template("index.html")
```

A user goes to a website and
lands on the /login page.

A user goes to a website and
lands on the /login page.

GET

A user enters their username
and password and clicks Log In.

A user enters their username
and password and clicks Log In.

POST

A user submits a birthday.

A user submits a birthday.

POST

A user navigates to the main page of the Birthdays website.

A user navigates to the main page of the Birthdays website.

POST

Forms

Form element

Route to request

Request method



```
<form action="/" method="post">
```

...

```
</form>
```

Elements of forms

```
<input name="friend" type="text">
```

```
<input name="month" type="number">
```

```
<input name="day" type="number">
```

```
<button type="submit">Submit</button>
```

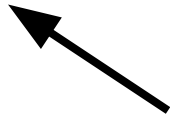
Elements of forms

```
<input name="friend" type="text">
```

```
<input name="month" type="number">
```

```
<input name="day" type="number">
```

```
<button type="submit">Submit</button>
```



Submits form when clicked!

Add a form

In **index.html**, add a form to submit new birthdays. The form should have the following attributes:

- The form should submit to the default "/" route, using the POST method.
- The form should include an **input** for a **name**, as well as the **month** and **day** of a birthday.
- The form should include a button to **submit** the form.

Updating a
model

```
request.form.get("friend")
```

```
<input name="friend" type="text">
```

```
INSERT INTO table (column1, column2)  
VALUES (value1, value2);
```

```
db.execute()
```

```
db.execute("SELECT * FROM birthdays  
          WHERE month = ?", month)
```

```
db.execute("SELECT * FROM birthdays  
          WHERE month = ?  
          AND day = ?", month, day)
```

Update a database

In **app.py**, insert a new entry to the **birthdays** table of **birthdays.db** when the user submits data via POST:

- Use **request.form.get** to get the values from each named input element.
- Use **db.execute** with **?** placeholders to insert those values into the **birthdays** table.
- Be sure to *validate* user input. For example, check if a value is **None** before inserting the value into the database.

Template Rendering

```
render_template("index.html",  
                message="Hello")
```

```
<p>The message is: {{ message }}</p>
```

```
render_template("index.html",  
                bdays=["Feb 2", "Feb 28"])
```

```
{% for bday in bdays %}  
    <p>{{ bday }}</p>  
{% endfor %}
```

```
render_template("index.html",  
                bday={"month": "Feb", "day": 24})
```

```
<p>This birthday is in {{ bday.month }}</p>
```

{% ... %}

Jinja expressions (e.g., **for**)

{{ ... }}

Jinja variables (e.g., **bday**, **message**)

Render birthdays

In **app.py** and **index.html**, query for birthdays and display them in a table.

- Use **db.execute** to query for a list of birthdays in the **birthdays** table.
- Use Jinja syntax to print out birthdays one by one in a table.

This is CS50

Week 9