

HTML Templates

The Problem

- We want to serve custom HTML
- You want to build a chat feature for your app
 - Users will submit their messages
 - Messages will appear to all users
 - Messages are contained in your HTML
 - How do we serve HTML that will change as users send messages?

HTML Templates

- Instead of writing complete HTML files
 - Write HTML templates
- An HTML template is an "incomplete" HTML file that is used to generate complete pages
- Use additional markup to add placeholders in the HTML
- Replace the placeholders with data at runtime

HTML Templates

- Example template with 3 placeholders
- The title, description, and image_filename will be replaced later
- Provide values for these 3 placeholders to serve a response

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>This page was generated from a template</title>
</head>
<body>

<h1>{{title}}</h1>

<p>{{description}}</p>



</body>
</html>
```

HTML Templates

- To substitute the placeholders
 - Use any string manipulation that gets the job done
 - Find/replace is the simplest solution
 - May want more advanced approaches if you want to add more functionality

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>This page was generated from a template</title>
</head>
<body>

<h1>{{title}}</h1>

<p>{{description}}</p>



</body>
</html>
```

Common Template Features

- Loops
- To add loops to your templates
 - Choose syntax for the start and end of the loop

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>This page was generated from a template</title>
</head>
<body>

  {{loop}}
  <h6>{{content_from_data_structure}}</h6>
  {{end_loop}}

</body>
</html>
```

Common Template Features

- Use string manipulation to find the start and end tags
- Iterate over your data
 - Add the contained HTML with the placeholder replaced for each value of your data

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>This page was generated from a template</title>
</head>
<body>

{{loop}}
<h6>{{content_from_data_structure}}</h6>
{{end_loop}}

</body>
</html>
```

Common Template Features

- Conditionals
- Can use similar approach as loops
- Choose syntax for the start and end of each block in the conditional

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>This page was generated from a template</title>
</head>
<body>

  {{if cookie_set}}
  <h6>Welcome back!</h6>
  {{else}}
  <h6>Welcome!</h6>
  {{end_if}}

</body>
</html>
```

Common Template Features

- Search for your tags
- Extract and evaluate the conditional
 - Choose how this will be evaluated
- Add the appropriate block of HTML to the page

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>This page was generated from a template</title>
</head>
<body>

  {{if cookie_set}}
  <h6>Welcome back!</h6>
  {{else}}
  <h6>Welcome!</h6>
  {{end_if}}

</body>
</html>
```

HW2 Templates

- You'll need to dynamically display uploaded images on your home page
- Add each image as the src in its own img element
 - The browser will request these images in separate HTTP requests using your objective 3 paths
- General strategy:
 - Write an HTML template with a loop containing an img element with a variable for the src attribute
 - Iterate over all the requested image names and use the filenames as the content for the loop
- The syntax for your template is up to you

Examples