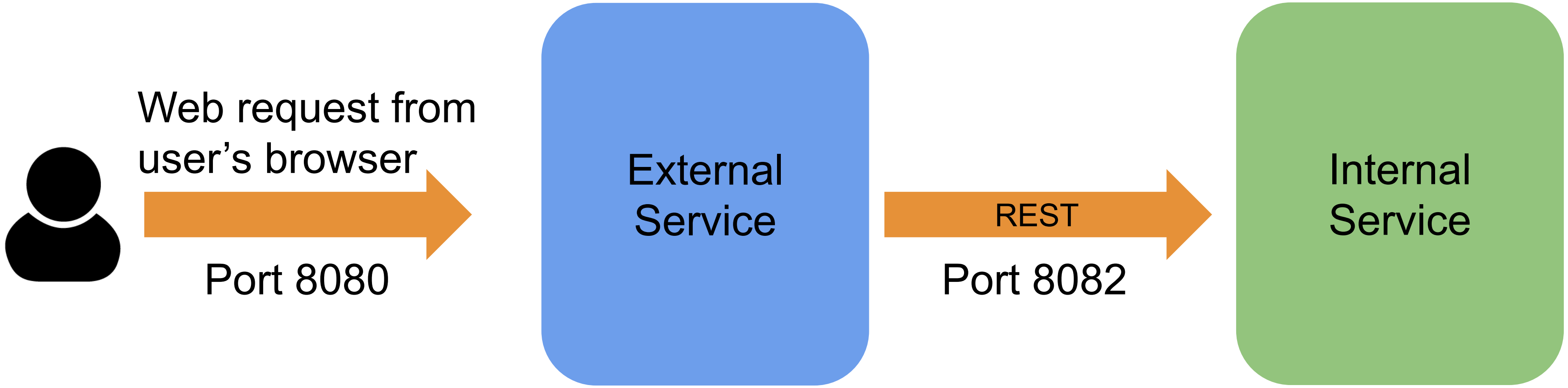
ROITRAINING
MAXIMIZE YOUR TRAINING INVESTMENT

Cheat Sheet:
Case Study Example Starting Point

About the Starting Point App

- The starting point app is provided as an example to help get started
    - Uses Node.js with the express web server on two microservices
- The *internal* service receives REST requests on port 8082 and returns mock data
- The *external* service inserts JSON from the server into an html template in the Views folder and returns it to the requestor on port 8080



Using the Startpoint

- In Cloud Shell, run the following commands:

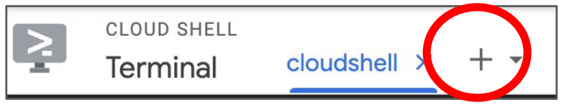
  -
    ```
    wget https://storage.googleapis.com/deloitte-training/events-app.zip
    ```

  -
    ```
    unzip events-app.zip
    ```

  -
    ```
    cd events-app/internal
    ```

  -
    ```
    npm install
    ```
    - This installs the application dependencies specified in package.json

  -
    ```
    node server.js
    ```
    - This starts the internal service
- Open a second Cloud Shell terminal by clicking the + and run these commands:

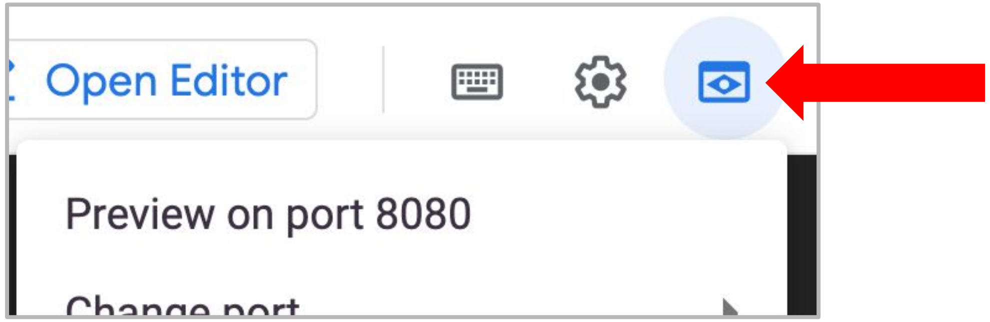
  -
    ```
    cd events-app/external
    ```

  -
    ```
    npm install
    ```

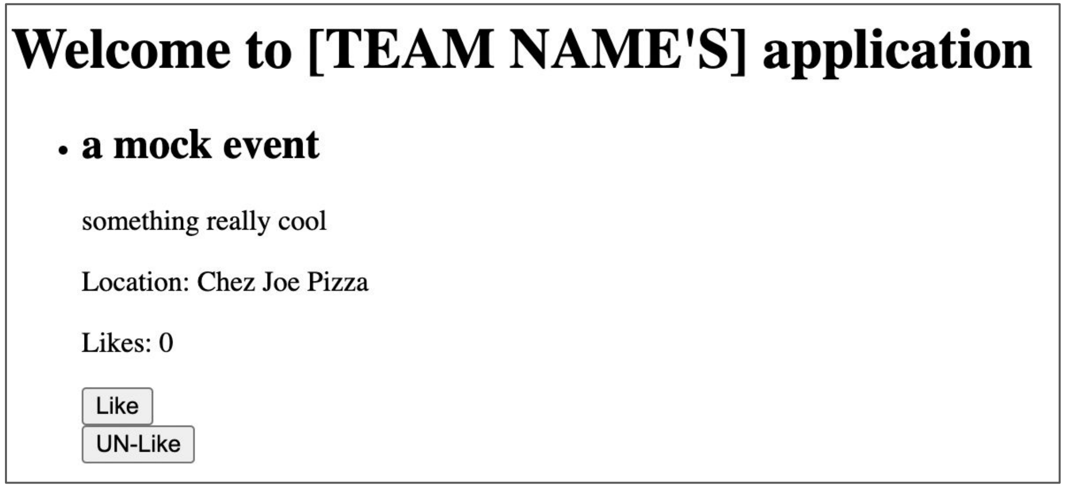
  -
    ```
    npm start
    ```
    - This starts the external service

Using the Startpoint

- Once both services are running in Cloud Shell, it can be tested with the **Preview on port 8080**



- **A new browser tab should open displaying the app startpoint**



Welcome to [TEAM NAME'S] application

- a mock event

something really cool

Location: Chez Joe Pizza

Likes: 0

Tip: Editing Files in CloudShell with nano

For Information only:

- Google Cloud Shell includes a built-in code editor
    - Easy to use
- You can also use **`vi`** or **`nano`** editor in Cloud Shell
    - You are welcome to use them if you know how
        - Beyond the scope of this course
- Here are a few tips on using nano if you want to try:
    - To edit a file, type: nano *`filename`*
    - Use the arrow keys on your keyboard to navigate within the file
    - To save the file and exit the editor:
        - Press **`<CTRL>+X`**, press **`Y`** and then press **`<Enter>`**

Customizing the App

- Once you get the app running, customize the app with your name:
  - In Cloud Shell Editor, use the tree view on the top-left to navigate the folders
  - Edit the **`events-app/external/views/layouts/default.hbs`** file
  - Change [TEAM NAME] to your name in two places
  - Save the file and reload the browser tab with the app preview
- Throughout the course, you will add additional features to the app
  - Add additional fields to the event
  - Ability to add images to events
  - Currently all data just saved in memory
    - We will add a database later
- Take a look at **home.hbs** in the **`events-app/external/views/`** folder

Dependencies

For information only:

- The external and internal services both use the following npm packages:
  - Express: a web server: https://www.npmjs.com/package/express
  - body-parser to convert JSON and form data in the request into parameters: https://www.npmjs.com/package/body-parser
  - Mocha, Chai, and SuperTest (for unit testing):
    - https://www.npmjs.com/package/mocha
    - https://www.npmjs.com/package/chai
    - https://www.npmjs.com/package/supertest
  - nyc for code coverage reporting: https://www.npmjs.com/package/nyc
  - The external service uses the following additional libraries:
  - express-handlebars (a templating library): https://github.com/ericf/express-handlebars
  - Nock (for mocking the api call): https://www.npmjs.com/package/nock

Stopping and Starting the Services

- If you need to stop a service, simply press `<CTRL>+C` in the Cloud Shell terminal
- To start the service again:
  - For internal, run `node server.js`
  - For external, run `npm start`

Implementing Twelve-Factor App

- After you get the case study running, as a group, come up with specific ways the case study application conforms to the twelve factors
    - There is a slide in your Google Slides document to record your results

Success!

- **Congratulations**! You have successfully run the class case study