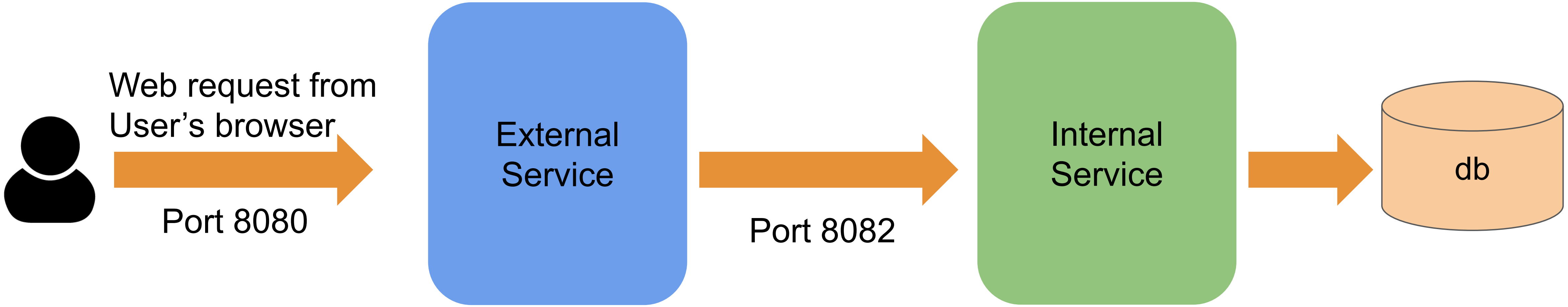
ROITRAINING
MAXIMIZE YOUR TRAINING INVESTMENT

Cheat Sheet:
Case Study Example Code

About the Case Study Example Code

- The Case Study Example Code is provided as an example app to get started
  - Uses Node.js with the express web server on two microservices
- The *internal* service receives REST requests and returns data from Firestore
- The *external* service receives web requests on port 8080, calls internal service, inserts JSON returned from the internal service into an HTML template in the Views folder and returns it to the requestor



Using Git (GitHub)

- The following slides provide steps on using GitHub
  - We will use GitHub for several activities
- We will use a personal GitHub account to do all classroom activities
  - This means a personal email is needed to set up a GitHub account if you don't already have one
  - If you have an existing personal account on GitHub, recover the password if you cannot remember current password
- Join GitHub now → https://github.com/

Using Git (GitHub) (continued)

- Create a GitHub personal access token:
  - From the https://github.com/ page, in the upper-right corner, click your profile photo, then click **Settings**
  - In the left sidebar, click **Developer settings**
  - In the left sidebar, click **Personal access tokens**, and then click **Tokens (classic)**
  - Click **Generate new token | Generate new token (classic)**
    - Provide a **Note** for your token of `classtoken`
    - Set the expiration to 7 days
    - Select the **Repo** scope
  - Click **Generate token**
- Copy the generated token and save it somewhere secure. **You cannot view it again**.
  - You will use the token as your GitHub password when using the **git** command
  - You will need it multiple times

Using Git (GitHub) (continued)

- Create a public repository called `events-app-internal` in your GitHub account
    - Do NOT add anything (e.g., a ReadMe)
    - Make a note of the repo’s https address (copy and save it somewhere)
- Create a second repository called `events-app-external` in your GitHub account
    - Do NOT add anything (e.g., a ReadMe)
    - Make a note of the repo’s https address (copy and save it somewhere)

Using Git (GitHub) (continued)

- Switch to the browser with Cloud Shell
  - Open a new Cloud Shell tab by clicking the + button + ▾
  - Execute the following commands:
    - `git config --global user.email "your_email_on_github"`
    - `git config --global user.name "your_github_user_name"`
    - `git config --global init.defaultBranch main`
    - `git config --global credential.helper store`
      - This will keep your token cached for subsequent activities
      - Will be destroyed with accounts at end of course

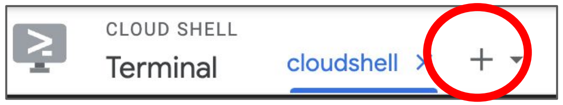
    - Verify with: `git config --global --list`

Using the Startpoint

- In the Cloud Shell terminal, run the following commands:
  - `git clone https://github.com/beachkodare/2023_10_24_phx_source_code.git sources`
  - `cd sources`
  - `mv example_application.zip ~`
  - `cd ~`
  - `unzip example_application.zip`
- Each directory now contains a microservice
- Build and run the “internal” service
  - `cd internal`
  - `npm install`
    - This installs the application dependencies specified in **package.json**
  - `node server.js`
    - This starts the internal service

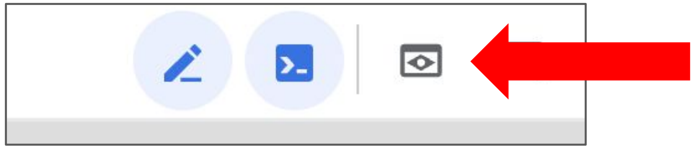
Using the Startpoint (continued)

- Open a second Cloud Shell terminal by clicking the + and run these commands:
  - `cd external`
  - `npm install`
  - `npm start`
    - This starts the external service
    - Notice you are running **npm start** and not **node server.js**
    - Look in the **package. json** file to see what **npm start** does

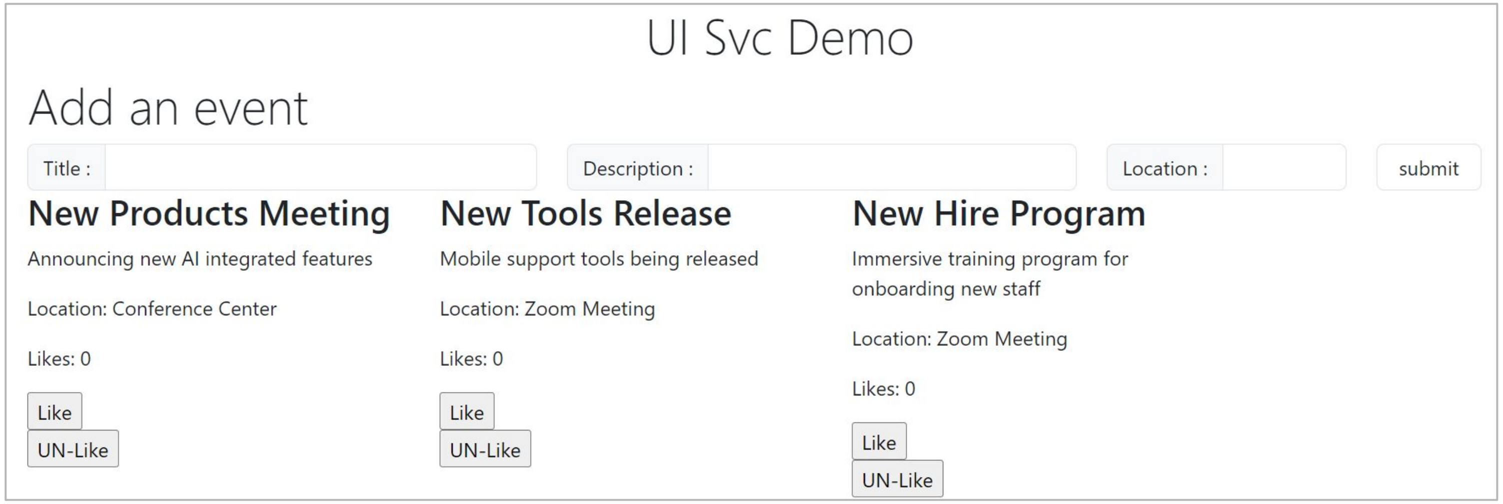


Using the Startpoint (continued)

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



- A new browser tab should open displaying the app startpoint with “fake” events



UI Svc Demo

Add an event

Title : | Description : | Location : | submit

New Products Meeting

Announcing new AI integrated features

Location: Conference Center

Likes: 0

Like
UN-Like

New Tools Release

Mobile support tools being released

Location: Zoom Meeting

Likes: 0

Like
UN-Like

New Hire Program

Immersive training program for onboarding new staff

Location: Zoom Meeting

Likes: 0

Like
UN-Like

- Fill out the “**Add an event**” form and click **Submit**
  - You should now see your event

Customizing the App (Optional)

- Once you get the app running, if you wish, customize the app by changing the **`external/views/layouts/default.hbs`** file
  - This is an express-handlebars template
- Throughout the course, you can add additional features to the app if you like **(but not required)**
  - Add a location to the event
  - Ability to “like” an event
  - Ability to add images to events
  - Etc.

Dependencies

- 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