

Hands-on: Developer Tools

Student worksheet · 15 minutes

You will explore OWASP Juice Shop with your browser's built-in developer tools.

Setup: get Juice Shop open

If your instructor gives you a link (a hosted instance), use that and skip to the next page. If no link is given, run Juice Shop on your own laptop:

1. **Install Docker** (Docker Desktop on Windows/Mac, Docker Engine on Linux) and make sure it is running.
2. Download the image:

   ```
   docker pull bkimminich/juice-shop
   ```

3. Start it (`NODE_ENV=unsafe` enables the challenges Juice Shop otherwise disables under Docker, such as Reflected XSS):

   ```
   docker run --rm -e NODE_ENV=unsafe -p 127.0.0.1:3000:3000 bkimminich/juice-shop
   ```

4. Wait about 30 seconds, then open `http://localhost:3000` in your browser. Close the welcome banner and the cookie notice.

Rule: do this only on our own Juice Shop. The same steps on a real website, without written permission, are an offence under the IT Act.

1. Open the developer tools

1. Open the tools: `F12`, or `Ctrl+Shift+I` (Windows/Linux), or `Cmd+Option+I` (Mac). In Firefox the keys are the same; some panels have other names (**Application** is **Storage**).
2. Click the **Elements** tab. Click the arrow/select icon (top-left of the tools), then click a product picture on the page. The tools jump to its `<img>` line in the page tree.
3. Double-click the product name in the tree and type something else. The page changes at once. Now reload: it is back.

Why: this tree is the **DOM**, your browser's copy of the page. You changed *your* copy only; the server did not change.

2. One search, one request

1. Click the **Network** tab. Tick **Preserve log**. Click the **Fetch/XHR** filter (Firefox: **XHR**).
2. Click the magnifier at the top of Juice Shop, type `apple`, and press Enter.
3. Find the request `search?q=` and click it. Look at:
   - **Headers**: method `GET`, path `/rest/products/search?q=`, status `200`.
   - **Response**: JSON data, starting `{"status":"success","data":[...]}`.

Why: the server sends *data* (JSON), not a new page. A script in your browser then shows the matching products. This is a single-page application.

If stuck: No request appears: the filter is wrong (click **All**), or you searched before opening Network. Search again.

3. Register, log in, and find your token

1. Click **Account** → **Login** → *Not yet a customer?*. Register with any made-up email (e.g. `you@class.local`) and a password of at least 5 characters. Pick any security question.
2. Log in with that account. In **Network**, click the `login` request (`POST /rest/user/login`). In **Response** you see `{"authentication":{"token":"eyJ0eXAi...", ...}}`.
3. Click **All** in the filter, click any later `/rest/` or `/api/` request, and in **Request Headers** find `Authorization: Bearer eyJ0eXAi...`.

Why: after you log in, the server gives you a **token**. Your browser sends it with every request. Whoever has this token *is* you, as far as the server can tell. Your password is not sent again.

If stuck: Email taken: change one letter. Password too short: use 5+ characters. Nothing in Network: turn on **Preserve log**, then log out and in again.

4. Where the browser keeps the token

1. Open **Application** (Firefox: **Storage**).
   - **Local Storage** → `http://localhost:3000`: key `token` (the same `eyJ0eXAi...`).
   - **Cookies**: a cookie `token`. Look at the **HttpOnly** column: it is *not* ticked.
2. Open the **Console** and type `document.cookie` and press Enter. Your token is printed.

Why: the token lives on *your* machine, so you can read, copy or delete it. The cookie is not **HttpOnly**, so any script in the page can read it too. That is what makes cross-site scripting dangerous, later today.

If stuck: Console refuses to paste: type `allow pasting`, press Enter, then paste. `document.cookie` is empty: you are not logged in.

5. Challenge: find the Score Board

Juice Shop has a hidden page that lists all its challenges. There is no link to it in the menu. Find it.

1. Hint: the pages (routes) of a single-page application are written in its JavaScript.
2. Open **Sources** (Firefox: **Debugger**). Press `Ctrl+Shift+F` (Mac: `Cmd+Option+F`) to search all files, and search for `score`. You will find `/score-board` inside `main.js`.
3. Open `http://localhost:3000/#/score-board` in the address bar. A green banner says the **Score Board** challenge is solved, and it now appears in the menu.

Why: nothing was hacked. The page was only *hidden*, and the whole map of the app was already in your browser. Keep this tab open: every challenge you solve today turns green here.

If stuck: Search finds nothing: reload the page, then search again. The address opens the shop home page: make sure you typed `#/score-board` with the `#/`.

What you used

Elements (the page tree), **Network** (every request and response), **Application** (what the browser stores), **Console** (run code in the page), **Sources** (all the page's code). An attacker uses the exact same tools.