

Debugging with **Chromium** Dev Tools

Vikram Nagulan
Software Engineer @ Tridium

The background of the right side of the slide is a dark blue, stylized city skyline at night. The buildings are silhouetted against a dark sky, with some windows glowing with light. The text 'NS2022' is prominently displayed in white, with 'ACCELERATING INNOVATION' in a smaller, light blue font below it.

NS2022
ACCELERATING INNOVATION

Let's get started

- Workbench Help -> Doc Developer
 - start by "Building Javascript Applications" for Niagara
- **grunt-init-niagara**
- **grunt-niagara** (v2.1.0 now available)
- <https://github.com/tridium>

Dev Modes

- moduledev
 - load source files directly from local disk instead of module jar files
 - speed up development
 - Needs "developer" feature added to your license
- webdev
 - load the un-minified original source and not the optimized minified one
 - aid in debugging

Web Dev Best Practices

- **grunt watch** - Always Be Testing
 - Test Driven Development (TDD)
- For JS, point Dev Tools at your tests, then station
- For CSS, point Dev Tools at station

Help -> Doc Developer -> Building Javascript Applications

Your Niagara station is started up and ready to run tests!
Every time you save
a file, linting and tests will be re-run.

To test in more browsers (like mobile devices), connect them
to the following

URL:

http://<Your IP>:9876/__karma/

Running "watchForChange" task

Waiting...



NS2022
ACCELERATING INNOVATION

Chromium Developer Tools

- Inspect Element
- Select Element
- Styles Tab

JavaScript Console

- Evaluate JavaScript
- `$0`
- Interact with widgets directly (like this one)
- `$($0).data('widget')`

JavaScript Debugger

- Step through code
- Break on errors (!!)
- Navigate async code
 - Turn on *niagara.useNativePromises*
- Save code "snippets" to run repeatedly.

Network tab

- Visualize network traffic
- See who's initiating network calls
- ...

Debugging Workbench

- Tools → Options → Web Browser → Web Development Tools
- Click the bug! 🐛
- Paste in Chrome
- See jxBrowser Spy Page to debug Px
- `niagara.webwidget.debug.delay`

Good to know

- "Enable Remote Logging" in DebugService
- "Show Stack Trace" in WebService

Thanks !!

- *developer.chrome.com/docs/devtools/*
- *Help -> Doc Developer -> Building Javascript Applications*



NS2022
ACCELERATING INNOVATION