Best and Worst Practices - DICOMweb™

Brad Genereaux
Agfa HealthCare
DICOM WG-27 Industry Co-Chair
555 Rule

- 5 seconds to find documentation
- 5 minutes to Hello World
- 5 hours to functional prototype
Supporting 5 Seconds to Documentation

• Put documentation in an easy-to-reference place
  – i.e., http://{server}/docs
• Implement Service Info transaction
Supporting 5 Minutes to Hello World

• Lay resources out in a logical way
• Use REST simulation tools to access resources
• Show in your documentation specific URLs to call
  – Suggest appropriate test data
Supporting 5 Hours to Working Prototype

- Use test automation to guarantee consistency
- Ensure compliance to standards
  - Participate in WG-31, Conformance
- Be vigilant, yet tolerant (adaptable)
- Use descriptive error codes and payloads
Security

• Consider security throughout the development process
  – Always use HTTPS, even in secured environments
  – See next presentation!
Eat your own dogfood

- In your own server/client applications, utilize the same API methods for communication rather than developing your own
- Stronger integration for you, and a battle-tested API for your consumers
General REST Best Practices

- Self-documenting
- Nouns in path, verbs by HTTP
- Complexity under the “?”
- Errors use HTTP error code mechanism

- As simple as possible, but no simpler
User ID / Password in URL

GET http://server.com/studies/
    ?00100010=DOE^JOHN&user=drjones&pwd=123456

• Supporting insecure (yet easy) practices sets a bad precedent, no matter how many times you say “don’t use in production”
Asking for too much

- Consider the following: Client needs to know patient position on objects within a series

- **WADO-RS call:**
  
  `{s}/studies/{studyUID}/series/{seriesUID}/metadata`

- **QIDO-RS call:**
  
  `{s}/studies/{studyUID}/series/{seriesUID}/instances/?includefield=00185100`
Incorrectly supporting errors

- Clients need to know what’s happening; make sure to use the right error code.
- Even worse, reporting OK when there was an error condition might mislead clients.
Avoid Proprietary Extensions

• Think it’s a good idea for adoption? Bring it to the standards organizations
  – If it is structural, bring it to DICOM WG-27
  – If it is profiling (i.e., workflow), bring it to IHE RAD

• If you’re going to do it anyway
  – Model (where possible) on existing standards
  – Refer to general REST best practices
Software Development Tools - DICOMweb™

Brad Genereaux
Agfa HealthCare
DICOM WG-27 Industry Co-Chair
Disclaimer

• Any tools I mention are not an official endorsement nor am I under any fiduciary relationship with any of these companies
Tools of the Web Developer

• Playgrounds
• Editors
• Scaffolding
• Builders
• Testing
• Other Useful Tools
Playgrounds

- Postman
- Curl
- JSFiddle
- CodePen
Editors

• Integrated Developer Environment (IDE)
  – Eclipse
  – Netbeans
  – IntelliJ
  – Visual Studio

• Lightweight Editors
  – Sublime
  – Notepad++
Scaffolding

- Tools that will lay out your project (i.e., directory structure).
  - Yeoman: Web applications
  - Maven: Java applications
  - Nuget: .Net applications

- Dependency management comes into play
  - Bower, NPM, require.js
Builders

• Tools that compile, test (via automation) and preview your application
  – Grunt
  – Gulp
  – Ant

• Many helpers run as part of the build process
  – JSLint / JSHint
  – Minify (JS / CSS)
  – Uglify
Testing

• Karma: Testing framework
  – Jasmine / Mocha / QUnit
• PhantomJS: Headless website testing
• Selenium: Browser automation
  – SauceLabs: Cloud-based browser testing
Other Useful Tools

- **Code repositories**
  - GitHub / Bitbucket
- **Virtual server environment**
  - Virtual Box
  - Docker, Vagrant
- **Troubleshooting**
  - Chrome development tools
  - Postman
  - Fiddler / Firebug
DICOMweb™ Specific Tools

• Available open-source server libraries
  – dcm4chee
  – Orthanc
• Available open-source client libraries
  – Cornerstone

Great resource: https://github.com/chafey/dicomWeb
DICOMweb™ Vagrants

- dcm4chee
  [https://github.com/mohannadhussain/dcm4chee4-quick-start-vagrant](https://github.com/mohannadhussain/dcm4chee4-quick-start-vagrant)

- Orthanc
  [https://github.com/chafey/orthanc-vagrant](https://github.com/chafey/orthanc-vagrant)