Trifolia-on-FHIR

Table of contents

Welcome

Welcome

Trifolia-on-FHIR is an editor for FHIR resources that uses a FHIR server natively as its back-end. All STU3-compliant FHIR servers work with Trifolia-on-FHIR.

Core Features

- Edit conformance resource types:
  - ImplementationGuide
  - StructureDefinition
  - ValueSet
  - CodeSystem
  - CapabilityStatement
  - OperationDefinition
  - Questionnaire
- Import and view any other resource in FHIR specification (such as Patient, Observation, MedicationStatement, etc.)
- Validate with the "Validation" tab on every resource editing screen, which uses FHIR.js.
- Export:
  - Implementation Guides and associated resources
  - Bundles
  - FHIR IG Publisher packages
- Import any resource or transaction bundle.

What's New

Login

Login

Click the "Login" button on the top right side of the screen. Trifolia-on-FHIR (ToF) re-directs users to identity/configure users' profile details. If you do not already have an account, you may register via this screen. Once you have registered and logged-in with the identity provider, your browser will redirect to the ToF homepage.

If this is your first time logging-in to ToF, you must create a Practitioner resource for ToF to identify you as the author of resources and associate your Practitioner with audit records when changing resources. If you configure ToF with multiple FHIR servers, you may need to create a new Practitioner resource for each FHIR server.

Users can click on their name at the top right side of the screen to further edit the Practitioner resource on the selected FHIR server.

Navigation

Navigation

The main navigation bar is on the top of the screen. Some menu items are hidden pending log-in.

- Home - This is the first screen users see after login. It presents high-level information about ToF.
- Browse/Edit - Search, select, delete, and create new resources depending on the resource type selected in the sub-menu.
  - Implementation Guides

    - Profiles
    - Capability Statements
    - Operation Definitions
    - Value Sets
    - Code Systems
    - Questionnaires
- Export - Export implementation guides from ToF in various formats (i.e., bundles, HTML).
- Import - Import resources from other locations into ToF.

On the right-side of the navigation menu, users will find:

- Drop-down menu - Select FHIR servers
- Edit Practitioner/Profile - Edit the Practitioner resource associated with your account.
- Logout - Logs your account off ToF
- Open from Computer - Appear as an "upload" icon. Users can open a resource directly from their computers, either an XML file or JSON file, and edit the resource in ToF without saving the resource to the FHIR server. When saving, the browser prompts users to re-download the updated resource as an XML or JSON file depending on the format when opened.
- Help - Shown as a question-mark icon, which opens this help documentation in a new window.

FHIR Versions

Depending on the FHIR server and the version it supports, the screens for editing resources may appear (e.g., STU3 vs. R4). The screens in ToF reflect the changes between STU3 and R4 resources.In STU3, for example, ImplementationGuide has "packages" with "resources" inside each package. In R4, ImplementationGuide has "resources" parallel to "packages," and each resource *references* a package.

FHIR Versions

FHIR Versions

Trifolia-on-FHIR supports multiple versions of the FHIR standard. ToF currently supports STU 3 and R4. Users can select a FHIR server with the drop down menu at the top right of every screen.

Process

Process

The work-flow for authoring an Implementation Guide:

1. Create an Implementation Guide
2. Create Profiles (StructureDefinition resources):
    - Create other resources (e.g., OperationDefinition, CapabilityStatement, ValueSet and CodeSystem) as needed for the profiles and implementation guide
    - Create samples of the profiles manually and import into ToF
3. Associate profiles, other conformance resources, and samples with the Implementation Guide
4. Export Implementation Guide using FHIR IG Publisher:
    - View the results of the export via the FHIR IG Publisher on the "Browse Implementation Guides" screen
    - The results include a Q/A tab, which identifies all errors the FHIR IG Publisher found during publication. Users should fix errors, when possible, and re-execute the export with the FHIR IG Publisher.
    - Users can download or upload the exported package to the appropriate GitHub repository for the Implementation Guide project.

Guidelines and Best Practices

Guidelines and Best Practices

Trifolia-on-FHIR has the functionality to allow users to completely customize resources. By following these guidelines, users can ensure the FHIR publisher successfully processes the Implementation Guide.

Implementation Guide

- The URL of the Implementation Guide must be in the format of `http[s]://xxx.yyy/zzz/aaa/ImplementationGuide/my-ig-id`. For example: `http://myproject.com/someRoot/ImplementationGuide/myproject-ig`
- The "id" of the implementation guide must align with the URL of the implementation guide. For example: If the URL of your implementation guide is `http://myproject.com/someRoot/ImplementationGuide/myproject-ig`, the id must be "`myproject-ig`". Users can select the "Change this resource's ID" button on the "Browse Implementation Guide" screen.
- The Implementation Guide should have a description. The main screen of the FHIR IG Publisher export displays the description.
- All contacts in the Implementation Guide appear as authors in the FHIR IG Publisher export.
- ToF only exports resources referenced directly within the Implementation Guide resource. Confirm the Implementation Guide resource references all resources.

All Resources

All resources within an Implementation Guide need URLs in one format. Based on the example above, if your implementation guide's URL is `http://myproject.com/someRoot/ImplementationGuide/myproject-ig` then all profiles (StructureDefinition resources) within the Implementation Guide must have URLs that start with `http://myproject.com/someRoot/StructureDefinition/`.

System Requirements

System Requirements

Users will need a modern browser (e.g., Chrome, Firefox, Internet Explorer, Safari) to use Trifolia-on-FHIR.

Administrators must ensure the following requirements to install Trifolia-on-FHIR in their individual servers:

- Windows *or* Linux
- FHIR Server (STU3 or R4)
    - Must support creating resources via a PUT with an ID
    - Must support the $validate operation

Help

Help

The help documentation is available in several formats:

- CHM
- DOCX
- PDF
- EPub

Export

Select Export in the tabbed tool bar on the top of the screen.
The Export page contains form fields that allow users to specify the details of their exports. Users can export the Implementation Guides (IGs) saved under the Browse/Edit tab at the top right side of the screen.

Users can export IGs as bundles or HTML with the IG Publisher. Once the form fields are complete, select the Export button on the left side of the scrolling tab at the bottom of the screen.

Bundle

Bundle exports produce a single download (pretty quickly) as a single XML file. This XML file is a FHIR Bundle that can be used to import the resources for the implementation guide in another FHIR environment.

HTML (and FHIR IG Publisher)

HTML exports produce a package (ZIP file) for use with the FHIR IG Publisher. Depending on whether you select "Run IG Publisher", the IG Publisher will automatically be executed for the package, and the output from the IG Publisher will be included in the download. If you select *not* to execute the IG Publisher, the package will still be produced and can be downloaded.

- The Export tool will take a few minutes to process. The length of time is correlated with the size of the export. Users will see the tool processing the export as lines of code executions. After completing the process, the export will automatically download to users' computers in a compressed folder.
- When the IG Publisher is executed, the output from the IG Publisher is copied to a public location in Trifolia-on-FHIR for preview.

GitHub

Exporting to GitHub will upload the resources within the selected Implementation Guide to the destination repository on GitHub. Resources that already exist at the specified paths will be overwritten with the content in ToF.
To export the Implementation Guide resources:

1. Navigate to the "Export" screen
2. Select the implementation guide to export
3. Select "GitHub" for the export format
4. If you are not already logged into GitHub within ToF, you will be prompted to login.
5. Specify a "Commit Message". This message is used by GitHub to describe the changes.
6. All resources within the implementation guide are listed in a table. Only resources that have a repository, branch and path specified for them will be uploaded to GitHub. To change the repository/branch/path for a resource
    1. Place a check-mark next to each of the resources you want to change
    2. Select "Change Selected" in the top-right corner of the table
    3. Select a GitHub repository
    4. A branch will automatically be selected if the repository has a default branch. If no default branch is specified, you will have to manually select the branch.
    5. Select the format that the resources should be stored in GitHub (JSON or XML)
    6. Indicate if the resources should be stored in a directory per resource type (for example, if you select the "resources" directory from the tree, but specify "Yes" to "Directory per resource type", the resource will be stored in "/resources/implementationguide/XXX.xml" (assuming the resource is an "ImplementationGuide" resource type)
    7. Select the destination directory to store the resources from the tree on the right. Optionally, you may create a new folder by typing the new folder name in the text field above the tree, and selecting "Add folder below selection".

See GitHub Integration for more information.

Import

ToF allows users to import resource files, copy/pasted resource JSON/XML, value sets from excel files, GitHub and VSAC content.
File imports allow users to drag-and-drop resources (e.g., StructureDefinitions, ValueSets, CodeSystems) from users' hard drives to the ToF tool.

Import Files

Users who upload more than 20 resources at once may experience a timeout error notification. In the event of a timeout error notification, users should reduce the size of the resource import.
Users can edit resource numbers based on individual needs.

Excel

Import value sets and code systems using the excel file import format. Using the "Import Files" tab, select (or drag-and-drop) an excel file that is prepared with the following structure:

- Worksheet: (name does not matter - the first worksheet in the workbook is used)
  - Column: "ID" (optional)
    - The column must exist, but does not have to have a value
    - This is the ID of the value set. It will be used to either update an existing value set or create a new value set with a pre-defined ID (assuming the FHIR server supports that functionality).
  - Column: "Name"
    - The name of the value set.
    - This name should be repeated for each code/row in the value set
    - Mapping: ValueSet.name
  - Column: "URL" - the URL for the value set
    - The URL of the value set that will be used to find or create the value set
    - This URL should be repeated for each code/row in the value set
    - Mapping: ValueSet.url
  - Column: "Code"
    - Mapping: ValueSet.compose.include.concept.code
  - Column: "Display"
    - Mapping: ValueSet.compose.include.concept.display
  - Column: "System"
    - This is the URL of the code system that the code belongs to
    - Mapping: ValueSet.compose.include.system

Additional

- The worksheet must have a header. ToF always ignores the first row.
- The heading of the columns do not matter, but the columns must appear in the order described above.
- If the value set already exists on the server, the entire value set will be overwritten with the data from the import excel file; including all codes in the existing value set that are not in the excel file.

Example

ID	Name	URL	Code	Display	System
vs1-test	Value Set 1	http://test.com/ValueSet/v1	asdf1	TEST1	http://test.com/CodeSystem/cs1
vs1-test	Value Set 1	http://test.com/ValueSet/v1	asdf2	TEST2	http://test.com/CodeSystem/cs2
vs2-test	Value Set 2	http://test.com/ValueSet/v2	asdf1	TEST1	http://test.com/CodeSystem/cs2

VSAC

VSAC imports require users' VSAC credentials, which are not persisted on the ToF server. If users select 'Remember VSAC Credentials,' the tool will store this information as cookies in the users' browser.

GitHub

To import resources from a GitHub repository, navigate to the "GitHub" tab in the "Import" screen. As soon as the tab is selected, you will be prompted to login to GitHub (unless you have already logged into GitHub in ToF previously).

1. Select a repository within GitHub that you have access to
2. The default branch for the repository it will be automatically selected. If the default branch does not exist or is not configured in the repository, you will have to select a branch manually.
3. A tree of all files available within the GitHub repository will be shown. Place a check-mark next to each of the files you wish to import.
4. The "Import" button is only enabled when at least one valid JSON or XML file is selected within the GitHub repository.
5. Select the "Import" button, and the resources will be imported from GitHub. If the resource already

exists in ToF with the same ID, the resource in ToF will be overwritten with the contents of the resource from GitHub.

See GitHub Integration for more information.

GitHub Integration

Authentication

The import and export screens both contain options for GitHub. As soon as the GitHub option is selected in either screen, you are prompted to login with your GitHub credentials. Once logged in, your GitHub authentication token is stored in cookies so that you do not have to login every time you select "GitHub" under the import/export screens.

After you have logged into GitHub, a GitHub icon appears in the top-right corner of the all screens. When clicked, this icon logs you out of GitHub within ToF.

ToF uses a pop-up window to authenticate with GitHub. If your browser blocks the pop-up window, ToF will not be able to authenticate with GitHub and you will receive an error.

Work Flow

The work flow within ToF for GitHub is to

1. Import resources from a GitHub repository into the selected FHIR server
2. Edit the resources using ToF
3. Export the resources back to the GitHub repository after they have the desired changes

Extensions

When importing resources, two extensions are added to each resource representing the location within GitHub for where the resource came from. This enables ToF to know where in GItHub to export the resources back to.

If you are exporting *new* resources to GitHub, these extensions will not yet exist and you will need to specify where the resources should be stored during the export (which will create the two extensions on the resource).

Glossary

Acronym	Definition
ToF	Trifolia-on-FHIR
FHIR	Fast Healthcare Interoperability Resources
VSAC	Value Set Authority Center