Trifolia-on-FHIR

Table of contents

Welcome

Trifolia-on-FHIR is a FHIR resource editor 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(Profiles/Extensions)
  - ValueSet
  - CodeSystem
  - CapabilityStatement
  - OperationDefinition
  - Questionnaire
- Import and view any resource in the FHIR specification (e.g., Observation, MedicationStatement).
- Validate any resource using the "Validation" tab editing screen, which uses FHIR.js.
- Export:
  - Implementation Guides and associated resources
  - Bundles
  - FHIR IG Publisher packages
- Import resources or transaction bundles.

Requesting Support

- Support requests are captured using JIRA Service Desk, located here: https://trifolia.atlassian.net/servicedesk/customer/portal/3.
- JIRA will require you log-in with an Atlassian account before you can submit or review support requests. . If you have not registered with Atlassian, you will be prompted to do so, first.
- Additionally, the FHIR Zulip chat has a channel dedicated for Trifolia-on-FHIR questions and announcements, located on chat.fhir.org in the #trifolia-on-fhir channel.

What's New

Release 1.6.0

New Publication Template

HL7 is developing an official style/look/feel/brand/template for implementation guides. ToF now supports this new template. Using the new template, you will see a more consistent look and feel to published implementation guides, as well as a simplified export format. The export no longer has a "framework" folder to contain 300+ files.

The new template also provides some flexibility with the top (red) navigation bar. Customize the navigation of the implementation guide when editing a page in the implementation guide by selecting "Show on top nav menu" and indicating the name of the menu you want the page to appear under. See Narrative Pages for more information.

The new template automatically generates a table of contents and an "Artifact Index" page.

Permissions for Implementation Guides

This new option copies permissions for an implementation guide to resources within the implementation guide. This functionality is primarily for cases where the permissions to the implementation guide have changed; and those changes need to be propagated down to the resources within the implementation guide.

You cannot change permissions for a resource you don't already have access to. If you don't have access to one or more of the resources within the IG, you cannot copy permissions from the IG to those resources. See Security and Permissions for more information.

Admin Role

For those who have their own installation of ToF: The new admin role designates users as administrators of the system with full control over ToF. Admins can list users who have logged into ToF and broadcast messages to users currently logged in. This update allows for additional administrative functionality (e.g., reporting) in the future.

Publish Queue

As ToF gets more popular, more people are publishing their implementation guides at the same time. We've noticed that servers can get taxed by the resource-intensive "Publish" requests. We implemented a queue, which allows administrators to configure the maximum number of publication requests that can run in parallel. If, for example, an installation is configured to run two publication requests in parallel, anyone else that makes a request to publish will enter the queue. Once one of the currently-running publish requests is complete, the next request in the queue will begin publishing.

Development Log

New Feature

[TRIFFHIR-225] - Ability to search elements for content when editing a structure definition
[TRIFFHIR-228] - Allow searching by "ID" on the Browse Profiles screen.
[TRIFFHIR-257] - Queuing for implementation guide publishing
[TRIFFHIR-261] - Basic admin functionality to list users, active users, broadcast a message
[TRIFFHIR-262] - Use the new official FHIR template
[TRIFFHIR-294] - Add page title to previous/next links in the generated markdown pages
[TRIFFHIR-295] - Copy permissions from IG to child resources
[TRIFFHIR-296] - Support group resources in implementation guide

Defect

[TRIFFHIR-220] - Wrong default FHIR version
[TRIFFHIR-246] - ServiceRequest vs. ServiceDefinition for CapabilityStatement > REST > Resource > Type
[TRIFFHIR-256] - Media images should not be included in ImplementationGuide resource when exported
[TRIFFHIR-258] - Adding a new slice to an already sliced element removes slice's discriminator
[TRIFFHIR-291] - ToF not preserving XHTML lang tag
[TRIFFHIR-305] - Fix slow responding text fields

Improvement

[TRIFFHIR-195] - Align profile editing grid with FHIR specification's layout
[TRIFFHIR-227] - Change how user adds resources to an IG from the "Edit IG" screen
[TRIFFHIR-248] - Update references when changing the ID of a resource
[TRIFFHIR-260] - Support specifying identifiers on CodeSystem

Additional Help

Additional Formats

The help documentation is available in several formats:

- CHM
- DOCX
- PDF
- EPub

Real-time Introduction (Guided Tour)

Click the "i" (information) button in the top-right corner of ToF to begin a real-time introduction to the screen. This introduction will walk you through the important elements of your screen and present a brief description of each element. If you open multiple tabs, the introduction will only describe the elements in the current tab.

Login

Login

Click the "Login" button on the top right side of the screen. Trifolia-on-FHIR (ToF) re-directs users to the identity provider to either register or login. 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 profile (which is represented using a Practitioner resource in ToF's FHIR server) for ToF to identify you as the author of resources and associate your profile with audit records when changing resources. If you configure ToF with multiple FHIR servers, you may need to create a new profile for each FHIR server you select.

Users can click on their name at the top right side of the screen to further edit the profile for their user 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.

- File
  - Home - This is the first screen users see after login. It presents high-level information about ToF.
  - Open from computer - 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.
  - Documentation - Opens this help documentation in HTML format.
  - Request Support - Opens the support page for ToF, where users can submit support requests (defects, new ideas for features/improvements and general questions).
  - Settings - This opens the settings window for 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
- Import - Import resources from other locations into ToF.
- Export - Export implementation guides from ToF in various formats (i.e., bundles, FHIR IG Publisher package, GitHub, etc).
- Publish - Publish your implementation guide using the FHIR IG Publisher.

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

- A label indicating the currently selected FHIR server. You may click on this label to open the settings for ToF and select a different FHIR server. This is the same as clicking on the File > Settings menu.
- A label indicating the user that is currently logged in. You may click on your name to edit your profile. A user is represented as a FHIR Practitioner resource.
- An icon for logging out of GitHub (if you are logged into GitHub within ToF).

- An icon for logging out of ToF (if you are logged into ToF).
- An icon for getting real-time guidance on how to use each screen. When clicked, a tour of the page will be started, highlighting key points of interest. *Note: not all screens support "tour" functionality.*

FHIR Versions in the UI

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.

Getting Started

IG Design Overview

1. Create/Open an ImplementationGuide (Project).
2. Create ValueSets and CodeSystems in the UI.
3. Create profiles (StructureDefinition resources) in UI. Create elements and constrain them to value sets created in step #2.
4. Import images into the IG to insert into narrative.
5. Create narrative pages for the IG.
6. Mark example resources as “example” with optional subject/context.
7. Publish the IG.
8. Review the published IG.
9. Review the QA results and resolve errors.
10. Export as HTML package and store in GitHub.

Create Implementation Guide

1. Open and log into Trifolia-on-FHIR.
2. Select File > New Project.
3. Specify an ID that is unique to your IG (e.g., test-ig).
4. Specify a URL unique to your IG (e.g., http://test.com/fhir/implementationguide/test-ig). The last leaf/level of the URL should match the ID to prevent errors during publishing.
5. Specify a name for your IG. This is the computable format of the name, used similarly (but not the same) as the ID.
6. Specify a title for your IG. This is what is displayed to readers of the IG.
7. Specify a Package ID. If developing an HL7 IG, you will receive this information pending PSS approval.
8. Define permissions using the “Permissions” tab.
9. Save the IG. It is now set as the project you are working on.

Create Terminology

Create a new value set or code system:

1. Select Browse/Edit > Value Sets and/or Code Systems.
2. Select the blue “+” button in the top-right to create a new value set or code system.
3. Specify the details of the value set and/or code system.
4. Save the value set or code system. The value set or code system is automatically associated with your IG.

Locate existing value sets or code systems:

1. Select Browse/Edit > Edit Implementation Guide.
2. Select the “Resources” tab.
3. Select the “hand/pointer” icon in the top-right to add an existing resource.
4. Un-check “Showing resources for the “Test IG” implementation guide.”
5. Filter/Find then select the value set or code system.

Import a ValueSet from VSAC:

1. Select “Import” on the top navigation bar
2. Select the “VSAC” tab
3. Specify your VSAC credentials
   Your credentials are never sent to ToF servers, they are used only by your browser to communicate with

VSAC; they do not leave your computer, aside from being sent to VSAC for authentication.
4. Specify the OID for the value set you would like to import.
5. Click “Import”.
6. Assuming the value set is found, it will be imported and automatically associated with your IG.

Create Profiles

1. Select Browse/Edit > Profiles/Extensions.
2. Click the blue “+” icon in the top-right of the screen.
3. Indicate the URL for the profile. The beginning of the URL should match the IG’s URL (e.g., http://test.com/fhir/structuredefinition/my-profile if the IG is http://test.com/fhir/implementationguide/test-ig).
4. Verify the ID matches the end of the URL. This should auto-populate when you enter the URL (e.g., “my-profile” for the above URL example)
5. Specify a name and title, which the published IG will display.
6. Indicate the “type” of resource this profile constrains (e.g., “Patient”).
7. You may want to build your profile based on another profile. Specify a “Base Profile” that exists in the system.
8. Click “Save.”
9. Click the “Elements” tab.
10. Select an element to constrain.
11. Click “Constrain Element” in the right panel.
12. Re-define the element in the right panel to specify the difference between the element and the base definition.
13. Repeat for all constraints.
14. Define a description for the profile, which the published IG will display to the reader.
15. Save the profile.

Import Images

1. Click the “Import” menu from the top navigation menu.
2. Drag-and-drop image files from your computer to the import window.
3. Click “Import.”

Create Narrative Pages

1. Click Browse/Edit > Edit Implementation Guide.
2. Select the “Narrative/Pages” tab.
3. Provide a high-level description of the implementation guide, if desired. If no narrative pages are specified (see Step 4), then the IG’s home page will use this description.
4. Scroll to the bottom and click the “Add Narrative Page” button. This will create an index page that represents the home page for your IG. When you edit this page, ToF can automatically generate the content for the home page (index) based on the description of the IG. You can also specify custom content displayed on the home page.
5. Click the “+” on the index/home page to create child pages.
6. The “Downloads” template is available to add to your IG and customize, which will provide default content for a “Downloads” page.
7. Select “Show on top navigation bar” and specify a navigation menu name for the pages you want to appear on the top menu of the published IG.
8. When editing a page, you can click the icon that represents multiple images to insert imports.

Identify Examples

1. Click Browse/Edit > Edit Implementation Guide.
2. Select the “Resources” tab.
3. Click “Edit” next to an example resource.
4. Select “Example = Yes” or select a profile to set an example.
5. Click “Done.”
6. “Save” the IG after updating resources.

Publish

1. Click the “Publish” menu from the top navigation menu.
2. Select the options for your publication.
3. HL7 requires the latest IG publisher for ballots. The IG publisher updates occasionally include bugs, which leads to errors. You can select “No” pending resolution.
4. Selecting “No” for “Use terminology server” can quicken the publish execution, assuming you’re not worried about terminology.
5. Indicating “Yes” for “Download?” will offer the entire published package as a download to your computer when complete.
6. Click “Publish.” This process can be time-consuming, depending on the size of your IG. Please be patient. Once published, you can view the IG with the link on the status screen. Alternatively, you can select Browse/Edit > View Implementation Guide.

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

An implementation guide (IG) is a set of rules about how FHIR resources are used (or should be used) to solve a particular problem, with associated documentation to support and clarify the usage. Classically, FHIR implementation guides are published on the web after they are generated using the FHIR Implementation Guide Publisher.

The ImplementationGuide resource is a single resource that defines the logical content of the IG, along with the important entry pages into the publication, so that the logical package that the IG represents, so that the contents are computable. In particular, validators are able to use the ImplementationGuide resource to validate content against the implementation guide as a whole. The significant conformance expectation introduced by the ImplementationGuide resource is the idea of Default Profiles. Implementations may conform to multiple implementation guides at once, but this requires that the implementation guides are compatible.

- 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.

Resources

The FHIR specification defines a set of resources, and an infrastructure for handling resources. In order to use FHIR to create solutions for integration requirements, implementers must map their problems to resources and their content.

- 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/`.

Structure Definitions(Profiles)

A profile is a definition of a FHIR structure and is used to describe the underlying resources, data types defined in FHIR, and also for describing extensions and constraints on resources and data types.

The StructureDefinition resource describes a structure - a set of data element definitions, and their associated rules of usage. These structure definitions are used to describe both the content defined in the FHIR specification itself - Resources, data types, the underlying infrastructural types, and also are used to describe how these structures are used in implementations. This allows the definitions of the structures to be shared and published through repositories of structure definitions, compared with each other, and used as the basis for code, report and UI generation.

- The Name should be usable as an identifier for the module by machine processing applications such as code generation. The name must start with a capital letter, have at least two characters, and cannot contain spaces or special characters.
- When creating a new Structure Definition, user must specify the resource Type

Element Definition

- The profile editor loads the elements from the snapshot of the “base definition” profile.
  - If the base definition represents a profile that does not yet have a snapshot, it attempts to create a snapshot for the base definition profile.
  - If the base definition has other base definition profiles that are not in the system, the snapshot it returns will simply be the core FHIR profile for the “type” selected (ex: the underlying profile for “Observation”)
  - The “Base Definition” field should allow you to select pre-existing profiles that are based on a matching “type”, or the core FHIR specification’s profile *for* the underlying type.
- Cardinality
  - Users can set the cardinality of an element to an invalid value, but a Warning will appear in the Element Definition form and a Validation Error appears in the Validation tab if the user.
    - selects a min cardinality less than the base element's min cardinality.
    - selects a max cardinality greater than the base element's max cardinality.
- Maximum Field Length
  - maxLength can only be entered when the ElementDefinition.type is a primitive type, with the exception of “boolean” (instant time date dateTime decimal integer string uri base64Binary code id oid unsignedInt positiveInt markdown url canonical uuid).

Value sets

A ValueSet resource instances specifies a set of codes drawn from one or more code systems, intended for use in a particular context. Value sets link between CodeSystem definitions and their use in [coded elements](terminologies.html). When using value sets, proper differentiation between a code system and a value set is important. This is one very common area where significant clinical safety risks occur in practice.

- Value sets used by an implementation guide *should* have a "compose" defined which asserts either the enumerated codes that should be included in the value sets, or asserts other value sets that should be included (making the value set a "wrapper" of sorts).
- Enumerated codes are shown/edited in the "Compose" tab's "Concepts" section. This section is paged, showing five (5) codes at a time. You may search for a concept by either the code or display values by entering text in the "Code (search)" and "Display (search)" fields shown at the top of the table.
- Additional fields (such as the "Designations") may be modified for each concept by clicking the "Edit" button.

Below the table of (at most) five (5) concepts is a set of buttons which allow you to control which page you are viewing/editing.

- The << button returns you to the first page of concepts
- The >> button moves you to the last page of concepts
- The < button moves you one page backward
- The > button moves you one page forward
- Selecting a number will bring you to that specific page number

The "Value Sets" section allows you to indicate what value sets should be included in this value set. Each entry in the "Value Sets" section represents the canonical URL of the value set (ValueSet.url).
When publishing an implementation guide which has a value set that references other value sets, those *other* value sets must be available to the FHIR IG Publisher via one of the following methods:

- Included in the implementation guide itself (via a resource referenced in the ImplementationGuide resource)
- The ValueSet is publicly available by the URL of the value set (e.x. putting the URL of the value set in a browser should return the ValueSet in either XML or JSON format)
- The terminology server used by the FHIR IG Publisher (tx.fhir.org) has the value set pre-loaded

Binding values to elements in a profile

To create a fixed binding for an element in a profile:

Starting from the profile editor's "Elements" tab:

1. Select the element you want to bind the fixed value to.
2. Select the "binding" tab in the properties of the element (on the right side of the screen).
3. Check the checkbox next to "Fixed".
4. Select the corresponding "type" for the element (ex: "CodeableConcept" or "string" or whatever, depending on the data type of the element you are constraining).
5. Edit the value of the fixed binding according to what you want to make sure the implementer uses.
   In some cases, more complex data types (such as CodeableConcept) will show an "edit" (pencil) icon next the type you selected in #4 which opens a pop-up dialog box to have you enter the information.
   Types that are simple (such as "code" or "string") just simply show a text field for you to enter the value in.

Code Systems

The CodeSystem resource is used to declare the existence of and describe a code system or code system supplement and its key properties, and optionally define a part or all of its content.

- Code systems define which codes (symbols and/or expressions) exist, and how they are understood. Value sets select a set of codes from one or more code systems to specify which codes can be used in a particular context.
- The CodeSystem resource may list some or all of the concepts in the code system, along with their basic properties (code, display, definition), designations, and additional properties.
- Code System resources may also be used to define supplements, that extend an existing code system with additional designations and properties.

Narrative Pages

An implementation guide can (and should) contain narrative pages that explain the use-case for the implementation guide and any special requirements for the implementation guide. You can add these narrative pages in the "Edit Implementation Guide" screen under the "Narrative/Pages" tab.
The content of the pages is markdown, and the ToF user interface provides a WYSIWYG editor that is based on markdown syntax. The WYSIWYG editor supports basic markdown syntax, such as bold, italics, underline, headings, etc. More advanced markdown functionality (such as tables) may not display correctly in the editor, but will be supported during the publish process; even though the WYSIWYG editor doesn't show the tables, it may look like a table after you have published your implementation guide.

Navigation Menus

Each page in the implementation guide can be associated with a top-level navigation menu. Edit the page, and specify a name for the top-level navigation menu. When you edit additional pages and begin typing the same menu name, the already-existing menu name will be shown as a suggestion for selection. After publishing the implementation guide, each unique page name will show up as a top-level navigation menu and any pages that are associated with that menu will be shown as options within that menu.
The top navigation menu *does not* reflect the hierarchy of the pages, at all. The table of contents *does* reflect the hierarchy of the pages. The table of contents is automatically generated during the publish process, every time.

Images

To add one or more images to pages in your implementation guide, follow these steps:

1. Import your images via the "Import" screen. You may drag-and-drop the images into the "Import" screen's "File" tab.
   These images will be imported as "Media" resources. The "id" of the Media resource will be based on the filename of the image, and the exact filename will be stored as an "identifier" in the Media.
2. Ensure your newly imported Media resources are added to the IG's "resources".
   Make sure they are *not* marked as an example. Leave the "Example" field either "Undefined" or "No". Otherwise, your Media resources will be treated as an example and will be preserved during the implementation guide's export, which may produce errors during final publication.
3. Open the page(s) you would like the image to show in, place your cursor where you want the image inserted and select the "Insert image from predefined list" option in the Markdown editor.
4. Select the image you want to add.
5. Text will be placed at your cursor for the image you selected.

Note: The following image types are supported:

- .JPG
- .GIF
- .PNG
- .BMP

Technical Notes

When the IG is being exported or published, the pages in the IG have to be in a specific order. The first/root page of the IG *has* to be an index.html file. If the first page in an IG is *not* an index.html file, then ToF automatically adds one and makes the root page of the IG a child page of the new index.html page.
Because the ImplementationGuide resource does not have any properties for storing the content of each page, ToF creates contained Binary resources within the ImplementationGuide resource.

Export

Export

Select Export in the tabbed tool bar on the top of the screen.

in order to export artifacts, you must first select the Implementation Guide(IG).

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.

1. The "IG Publisher Package" generates a package using the published version of the IG to be used with the FHIR IG Publisher.
   - Users can select either JSON or XML format for the output
   - The "FHIR IG Publisher JAR" can be included in the export package
   - The Export tool will take a few minutes to process. The length of time is correlated with the size of the export.
   - After completing the process, the export will automatically download to users' computers in a compressed folder, and tthe user is prompted to download a ZIP package of all files necessary to execute the FHIR IG publisher.
   - When the IG Publisher is executed, the output from the IG Publisher is copied to a public location in Trifolia-on-FHIR for preview.
   - HTML exports produce a package (ZIP file) for use with the FHIR IG Publisher.

2. The "Bundle" export format is a Bundle of all resources referenced by the selected implementation guide, including the implementation guide resource
   - 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.
   - Users can select the type of output, JSON or XML, the bundle will be exported in.

3. The "GitHub" export format places all the resources within the IG into the specified GITHub repository/branch
   - Only resources that have a repository, branch and path will be exported to GitHub. At least one resource must have GitHub location information specified to export
   - If the file already exists in GitHub, it will be completely overwritten by this export
   - When this tab is selected from the Export page, the user will be prompted to enter their GitHub credentials. If the user does not log into GitHub initially, the "Login to GitHub" button can be selected before the export is commenced.
   - After a user has selected an Implementation Guide, the user must enter a commit message that will be associated with the IG on GitHub.

Import

Import

ToF allows users to import files, text, VSAC and GitHub content.

- **File** imports allow users to drag-and-drop FHIR resources (e.g., StructureDefinitions, ValueSets, CodeSystems) and Excel-based value sets directly from your computer to ToF.
  - Users can drag-and-drop files from their computers File Explorer directly into the Import screen, or select the "Click to Select" link to select the files they wish to import.
  - Once the user has selected the files in Explorer, the list of files to be imported will be displayed in the Import screen.
  - Users can click the **trash** icon to remove file(s) from the list of files to be imported.
  - Imported resources are sent directly to the ToF server as a transaction bundle. ToF displays an excerpt of the JSON or XML bundle of the file prior to uploading.
  - The **Results** tab displays the outcome of importing from the **Files** tab
- **Text** imports allows users to copy/paste JSON or XML content directly into Trifolia-on-FHIR to have it imported.
- **VSAC** imports allows users to import value sets and code systems directly from VSAC into ToF.
  - Imported value sets and code systems can be referenced by Structure Definition resources.

    - The VSAC mports 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.
- The **GitHub** tab allows users to import resources directly from GitHub into ToF.
    - After selecting the "GitHub" tab, the user will be prompted to login to GitHub using their GitHub credentials.

Note: 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 Value Sets

In the "Files" tab you may import excel spreadsheets (XLSX) that represent value sets. The spreadsheet document must have these columns, in this specific order:

- ID - This is the ID of the value set. This value in this column is repeated for each code/row. The value must be formatted as a valid ID. `Ex: 2.16.840.1.113883.42.3.1`
- Name - The name of the value set. This value in this column is repeated for each code/row.
- URL - The URL of the value set. This value in this column is repeated for each code/row. `Ex: http://some.com/test/fhir/valueset`
- Code - The code representing the concept. `Ex: 900000000000073002`
- Display - The display representing the concept. `Ex: "Sufficiently defined concept definition status"`
- System - The system URL that owns/maintains the code. `Ex: http://snomed.info/sct`

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).

Limitations

- Trifolia only exports the individual resources associated with the implementation guide, and does not include the entire IG Publication package. For example, the "framework" (html templates) folder is not included in the export.

- Trifolia only allows importing FHIR resources. Trifolia-on-FHIR allows the user to select any JSON or XML file from GitHub. If the user selects an XML or JSON file that is not a FHIR resource, the import will fail.
- GitHub does not allow retrieving/updating very large files. For example, if attempting to import/export a large ValueSet resource, GitHub may fail with a "Payload too large" error.

Signing Out

When you sign out of GitHub within Trifolia, this clears your GitHub session only within Trifolia. GitHub maintains its own session within your browser. To sign out of GitHub entirely, you will need to go to github.com and click "Sign out".

Validation

Validation

Trifolia-on-FHIR uses three validation methods to provide as much feedback to IG authors as possible:

1. **Real-time UI Validation**
   This validation checks the base FHIR specification requirements (i.e., cardinality, terminology bindings, value set requirements). This validation occurs as each field in ToF is changed to update and render to the user in real-time.
2. **Pre-publish Validation**
   When publishing an implementation guide from the "Publish" screen, the FHIR server's $validate operation is executed for each resource in the implementation guide. This is specific to the FHIR server in Trifolia-on-FHIR for this IG (e.g., HAPI, the FHIR server instance default).
3. **HL7 IG Publisher Validation**
   The FHIR IG Publisher executes this validation step automatically *during* the publishing process. Validation checks for relationships between all resources and pages within this IG package. This includes all applicable IG resources, profiles, extensions, value sets, etc. FHIR IG Publisher validation also validates HTML links within the package. You cannot execute this validation step externally/independently of the publish process for an entire IG.

Walk-through

The purpose of this page is to guide new users through Trifolia-on-FHIR:

1. Create account and Login
2. Select FHIR Release version (gear icon in top right)
3. Create new Implementation Guide
   - Option A: Create IG from scratch. Navigate to Browse Implementation Guides > click the "plus" + button at top IG list/table.
   - Option B: Import IG from a file. Import IG.xml from your computer ("Import" button at top and either drag-and-drop the IG.xml file into the "Files" tab or copy/paste the contents of IG.xml into the second tab).
4. Modify IG. Be sure to always *Save* (bottom left)
   NOTE: When creating and ordering pages, you can only pivot a page's position with pages that are siblings with one another. Pages that are children of the page you're trying to move or children of a page that's not the parent of the page being moved can not be swapped.
5. Create/import additional templates/profiles
   - Option A: Create Profile from scratch. Navigate to Browse Templates/Profiles > click the "plus" + button at top of Profile list/table.
   - Option B: Import profiles from directories on computer
6. Modify and constrain the templates/profiles to use case
7. Resolve all Validation errors and warnings on Validation (tab) within each profile
8. Export selected IG package. Suggested settings for initial export:
   1. Export Format: HTML (IG publisher)
   2. Run the IG Publisher: Yes

   3. Run the latest version of the IG Publisher: No
   4. Use terminology server: Yes/No (Suggest No if IG uses large standard codesets) Selecting Yes will verify applicable value sets and code systems externally
   5. Download: Yes
   6. Output format: XML
9. Confirm build logs against CI-publisher on Zulip > Notifications

Security and Permissions

Authentication

Trifolia-on-FHIR is designed to minimally require that the user authenticate in order to access the data that is stored on the FHIR servers that ToF is configured to use. Additional permissions may be required depending on the configuration of the ToF installation.

Permissions

If the ToF installation is configured to require permissions, only data that the user has been permitted to view/edit will be access to them in the user interface. The remainder of this section presumes that permissions are enabled in the installation.

Permissions are maintained for each individual resource in the system. For example, permissions may be different for an instance of an ImplementationGuide compared to a StructureDefinintion that the implementation guide references.

Each edit screen contains a "Permissions" tab which allows the user to define the permissions for the resource. The user may search for users and groups, and add read and/or write permissions to the resource for the selected users/groups.

The user may select a different resource to copy permissions from. This can be done either by:

1. Selecting a resource type and typing search criteria in the text field. Suggestions will be presented below the text field. Select one of the suggestions and press the "Copy" button.
2. Click the "Search" button next to the text field to select a resource using the advanced search pop-up window. Once a resource has been identified and selected, click the "Copy" button.

If you have been granted permissions to a resource via a group and that resource has other groups associated with it that you *aren't* a member of, the name of the group will not be shown and the "Permissions" tab will only show you the ID of those other groups.

If you do not have permissions to edit a resource, you will not be able to click the "Edit" button on the resource from the browse screen. Future enhancements may be made to allow the user to access the "Edit" screen in a disabled state when the user doesn't have edit permissions to the resource.

Managing Groups

All users may create/manage their own groups. A group may only have one manager.

To create/edit/delete groups, click your name in the top-right of ToF, and select the "Groups" tab. Changes made to the "Groups" tab are persisted immediately; pressing "Save" is not required and only applies to editing information for your profile.

When you create a group, you are automatically added as a member to the group. You cannot remove yourself as a member from the group.

Importing Resources

When importing *new* resources, the permissions for those new resources are defaulted to allow the user performing the import view/edit access. To allow additional permissions, you will need to edit each resource and grant additional permissions.

Implementation Guide Permissions

You may copy permissions for an implementation guide to resources within the implementation guide. This functionality is primarily for scenarios where the permissions to the implementation guide have changed, and those changes need to be propagated down to the resources within the implementation guide.

Note: You cannot change permissions for a resource you don't already have access to; if you don't have access to one or more of the resources within the IG, you won't be able to copy permissions from the IG to

that/those resources.
To do this:

1. Open a project
2. Select Browse > "Edit ImplementationGuide"
3. Select the "Permissions" tab
4. Click the "Copy" button in the "Copy IG Permissions" panel
5. When ToF is done copying permissions, you will be prompted to indicate how many resources were changed as part of the request

Glossary

Acronym	Definition
ToF	Trifolia-on-FHIR
FHIR	Fast Healthcare Interoperability Resources
VSAC	Value Set Authority Center
IG	Implementation Guide
VS	Value Set
CS	Code System

FAQ

General

My implementation guide has pages, but the table of contents is empty?
Make sure that the "Table of Contents" page's "Auto Generate Table of Contents?" field is set to "Yes".

Exporting/Publishing with the FHIR IG Publisher

The IG Publisher reports "Property name not found"
This may be due to dependencies being listed in the ImplementationGuide resource incorrectly.

Technical Details

System Requirements

System Requirements

End-Users

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

Administrators

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

- **Windows *or* Linux**
- **Java 8+** (to execute the publish process using the FHIR IG Publisher)
- **Jekyll** (to successfully complete the publish process using the FHIR IG Publisher)
- **FHIR Server** (STU3 or R4)
  - Must support creating resources via a PUT with an ID
  - Must support the $validate operation
  - Must support the $meta-delete operation

- Must support ImplementationGuide search query parameters:
  - resource
  - global
- Must support _has (reverse chaining) search criteria. For example: GET /StructureDefinition?_has:ImplementationGuide:resource:_id=<IG_ID>
- Must support _include search criteria to get a list of all resources related to an implementation guide. For example: GET /ImplementationGuide?_id=some-ig-id&_include=ImplementationGuide:resource&ImplementationGuide:global

FHIR Versions

FHIR Versions

Trifolia-on-FHIR supports multiple versions of the FHIR standard. ToF currently supports STU3 and R4. Users can select a FHIR server with the drop down menu at the top right of every screen.

REST API

Multiple FHIR servers

If the ToF installation is configured to support multiple FHIR servers, the first FHIR server is the default in the REST API. If you want to perform REST API operations on a different FHIR server, you must specify a **fhirserver** header in each request. The value of the **fhirserver** header must align with the **id** of one of the FHIR servers returnved by **/api/config**.

FHIR Server Proxy

An **/api/fhir** end-point in the API represents a "proxy" to the FHIR server(s) available within the ToF installation. The proxy endpoint supports GET, PUT, POST, DELETE requests for individual resources, as well as batches. Transactions are *not* supported by this proxy.

Authentication

Authentication is required to access the REST API. To get an authorization code to use with the REST API, login to the Trifolia-on-FHIR web application and open the settings screen, where the "Authorization Code" is displayed to you. Attach the authorization code in an "Authorization" header prefixed with "Bearer ". For example: "Authorization: Bearer XXXX" where XXXX is your authorization code.

Multiple FHIR Servers

The default FHIR server used for this proxy endpoint is the first FHIR server available in the drop-down of FHIR servers in the settings screen of the application.
If you do not want to use the default/first FHIR server, specify a "fhirserver" header in each request where the value of the header is the id of the FHIR server according to the settings screen in the application.
The settings screen shows you an example of a request using curl for whichever server you have selected.

Trifolia-on-FHIR Native API

Trifolia-on-FHIR's REST API is documented using Swagger. The publicly available installation of Trifolia-on-FHIR exposes custom API documentation at https://trifolia-fhir.lantanagroup.com/api-docs. The API described by **/api-docs** is the same API that the web application (user interface) uses.

Security and Permissions

Permissions for resources are stored in Resource.meta.security. A custom code is created for three types of permissions:

- Everyone - Anyone that has a user account in the installation.
- Group - One or more users (Practitioners) that are represented together as a single Group. Use a group to represent a team of users.
- User (Practitioner) - A single person that has access to the installation. ToF requires every user to

create a Practitioner that represents their user when the login and open ToF to a specific FHIR server for the first time.

There are two levels of permissions:

- Read - Allows the user to search/view the resource
- Write - Allows the user to update/delete the resource

WIth these concepts combined, the resource may have several security codes. For example:

```
{
  resourceType: "ImplementationGuide",
  meta: {
    security: [
      // Everyone has access to read/wite
      { system: "https://trifolia-fhir.../security", code: "everyone^read" },
      { system: "https://trifolia-fhir.../security", code: "everyone^write" },
      // Members of group test-group-id have access to read/write
      { system: "https://trifolia-fhir.../security", code: "group^test-group-id^read" },
      { system: "https://trifolia-fhir.../security", code: "group^test-group-id^write" },
      // A specific user (Practitioner) with id test-practitioner-id has access to read/write
      { system: "https://trifolia-fhir.../security", code: "user^test-practitioner-id^read" },
      { system: "https://trifolia-fhir.../security", code: "user^test-practitioner-id^write" }
    ]
  }
}
```

When a user searches for ImplementationGuide resources, ToF sends a search request to the FHIR server that includes a _security parameter with all possible variations that are applicable to the currently logged-in user. For example:

```
// un-encoded for readability
https://some-fhir-server.com/fhir/ImplementationGuide?_security=<system>|
everyone^read,<system>|group^test-group-id^read,<system>|user^test-practitioner-rid^read
// encoded
https://some-fhir-server.com/fhir/ImplementationGuide?_security=https%3A%2F%2Ftrifolia-fhir...%
2Fsecurity%7Ceveryone%5Eread%2Chttps%3A%2F%2Ftrifolia-fhir...%2Fsecurity%7Cgroup%
5Etest-group-id%5Eread%2Chttps%3A%2F%2Ftrifolia-fhir...%2Fsecurity%7Cuser%5Etest-
practitioner-rid%5Eread
```

When a user clicks the "Edit" button on a resource, this initiates getting a single/specific resource. The ToF server checks that the persisted resource grants the logged-in user permissions to view the resource before sending the resource back to the user's browser for viewing.

Similarly, when a user clicks "Save" or "Delete", the ToF server first retrieves the instance of the resource that is persisted on the FHIR server, checks whether the user has permissions to modify the resource, and rejects the request with a 401 Unauthorized response if the user does not have permissions. Otherwise, the resource is updated on the FHIR server according to the user's request.