Trifolia-on-FHIR

Table of contents

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.

Requesting Support

Support requests are captured using JIRA Service Desk, located here: https://trifolia.atlassian.net/servicedesk/customer/portal/3. JIRA will require that you be logged in with an Atlassian account before submitting/viewing support requests.

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 2.0.0 on ???

Permissions

Users can now manage permissions for resources, limiting who can see and/or edit resources. Each edit screen has a tab for "Permissions" which allows you to manage permissions for that resource. Permissions can be copied from other resources.

See the "Security and Permissions" section of the help documentation for more information.

Export and publish improvements

The functionality for export and publish have been separate into two separate screens. The "export" screen focuses only on exporting/downloading a .zip file of your IG from ToF, while the "publish" screen runs the FHIR IG Publisher against your implementation guide. The UI of the export screen has been modified to be more consistent with other screens.

Profile editor supports data-type constraints

We've added additional functionality to the profile editor that allows the user to drill deeper into the elements of a profile. This release allows the user to expand data-type elements and constrain more details about the profile.

Development Log

TRIFFHIR-126	New Feature	Export and publish improvements
TRIFFHIR-42	New Feature	Implement permissions management
TRIFFHIR-184	New Feature	Allow drill-down into data types
TRIFFHIR-62	Improvement	Allow user to publish from "View Implementation Guide" page
TRIFFHIR-153	Improvement	Create new profile - Type should not default to Account
TRIFFHIR-173	Improvement	Update publication framework to use title instead of name
TRIFFHIR-175	Improvement	Add .sh and .bat files when exporting with the IG publisher jar file
TRIFFHIR-186	Improvement	Allow user to cancel changes to an implementation guide's page
TRIFFHIR-196	Improvement	Include selected FHIR server in the browser's URL
TRIFFHIR-182	Improvement	Show elements from base profile
TRIFFHIR-171	Improvement	Improve readability/usability of "Type" drop down when creating new profile

TRIFFHIR-181	Improvement	Enabling a new item shouldn't collapse the right-hand menu
TRIFFHIR-177	Defect	Implementation Guide Validation/RAW not refreshing
TRIFFHIR-183	Defect	HTML Export not including one of the resources referenced by the IG
TRIFFHIR-194	Defect	Import fails in DEV without meaningful message to user

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.

- 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.

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/`.

Value sets

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

Adding images to pages

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. Add your newly imported Media resources 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 pre-defined 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

Export

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 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 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.

Import

Import

ToF allows users to import files, text, 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.

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.

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.

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 utilizes 3 validation methods to provide as much feed-back to IG authors as possible.

1. **Real-time UI validation while editing**
   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, allowing it 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 being utilized by ToF for this IG (ex: HAPI, or the FHIR server instance selected in the top right of ToF).
3. **HL7 IG Publisher validation**
   This validation is performed automatically *during* the publish process by the FHIR IG Publisher. 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 contained in the package. This validation method cannot be invoked externally/independently of initiating 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)
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.

Additional Help

Help

The help documentation is available in several formats:

- CHM
- DOCX
- PDF
- EPub

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

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
    - 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

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

Multiple FHIR servers

If the ToF installation is configured to support multiple FHIR servers, the first FHIR server is used by default in the REST API. If you wish to perform REST API operations on a FHIR server other than the first, you must specify a **fhirserver** header in each request. The value of the **fhirserver** header must be the **id** of one of the FHIR servers returned by **/api/config**.

FHIR Server Proxy

An **/api/fhir** end-point is available in the API that represents a "proxy" to the FHIR server(s) available within the ToF installation.

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.