TGM HIT protocol template

A protocol template for students of the HIT department at TGM Wien.

v0.2.0 November 25, 2025

https://github.com/TGM-HIT/typst-protocol

Simon Gao **Clemens Koza**

Contents

I Introduction

This template is aimed at students of the information technology department at the TGM technical secondary school in Vienna. It can be used both in the Typst app and using the CLI:

Using the Typst web app, you can create a project by e.g. using the "Create new project in app" button on the package's Universe page: https://typst.app/universe/package/tgm-hit-protocol.

To work locally, use the following command:

```bash
typst init @preview/tgm-hit-protocol
```

If you are getting started writing your protocol, you will likely be better off looking into the document created by the template: it contains instruction and examples on the most important features of this template. If you have not yet initialized the template, a rendered version is linked in the README. If you are new to Typst, also check out the Typst documentation: https://typst.app/docs/.

The rest of this manual documents the individual functions offered by this package. This is useful if you want to know what customization options are available, or you're not sure what parts of the template package do.

As a school-specific template, this package does not offer enough configurability to be used at different institutions. However, if you like this template, feel free to adapt the code (MIT-licensed) to your needs, or open a Github issue if you think the template could be adapted to work for your requirements.

II Module reference

II.a `tgm-hit-protocol`

The template's main module. All functions that need to be called are directly exported from this module.

- `template()`

```
template(
  subject: content string,
  course: content string,
  title: content string,
  subtitle: content string,
  author: content string,
  teacher: content string,
  version: content string,
  begin: datetime,
  finish: datetime,
  date: datetime,
  bibliography: content,
) -> function
```

The main template function. Your document will generally start with `#show: template(..)`, which it already does after initializing the template. Although all parameters are named, most of them are really mandatory. Parameters that are not given may result in missing content in places where it is not actually optional.

Parameters:

`subject` (`content` or `string` = `none`) – The subject, displayed on the title page, above the course.

`course` (`content` or `string` = `none`) – The course, displayed on the title page, above the title.

`title` (`content` or `string` = `none`) – The title, displayed on the title page.

`subtitle` (`content` or `string` = `none`) – The subtitle, displayed on the title, under the title.

`author` (`content` or `string` = `none`) – The author, displayed under the subtitle and in the footer.

`teacher` (`content` or `string` = `none`) – The name of the teacher, displayed on the title page.

`version` (`content` or `string` = `none`) – The version, displayed on the title page.

`begin` (`datetime` = `none`) – The begin date of the protocol.

`finish` (`datetime` = `none`) – The finish date of the protocol.

`date` (`datetime` = `datetime.today()`) – The current date, displayed on the title page and in the header.

`bibliography` (`content` = `none`) – The bibliography (`bibliography()`) to use for the thesis.

II.b `tgm-hit-protocol.glossary`

Wrappers for Glossarium functionality. The `glossary-entry()` and `register-glossary()` functions as well as Glossarium's `gls()` and `glspl()` are re-exported from the main module.

- `register-glossary()`
- `glossary-entry()`
- `print-glossary()`

```
register-glossary(..entries: arguments) -> content
```

Registers the passed entries with Glossarium.

Parameters:

`..entries` (`arguments`) – The positional-only entries for the glossary

```
glossary-entry(
  key: string,
  short: string,
  long: string content,
  description: string content,
  plural: string content,
  longplural: string content,
  group: string,
) -> dictionary
```

Stores a glossary entry for this thesis. One call to this function is equivalent to one array entry in Glossarium's `print-glossary()`'s main parameter.

Parameters:

`key` (`string`) – The key with which the glossary entry can be referenced; must be unique.

`short` (`string` = `none`) – Mandatory; the short form of the entry shown after the term has been first defined.

`long` (`string` or `content` = `none`) – The long form of the term, displayed in the glossary and on the first citation of the term.

`description` (`string` or `content` = `none`) – The description of the term.

`plural` (`string` or `content` = `none`) – The pluralized short form of the term.

`longplural` (`string` or `content` = `none`) – The pluralized long form of the term.

`group` (`string` = `none`) – The group the term belongs to. The terms are displayed by groups in the glossary.

```
print-glossary(title: content, ..args: arguments) -> content
```

Displays a glossary of the entries added via `glossary-entry()`.

Parameters:

`title` (`content` = `none`) – A (level 1) heading that titles this glossary. If the glossary is empty, the title is not shown.

`..args` (`arguments`) – Any extra parameters to the glossarium function of the same name.