Moldable Agent Harness

andrei.chis@feenk.com

Creating agents with structured output

Basic agent

```
agent := GtLAgent new.
agent markdownResponse.
```

```
agent sendMarkdown: 'Write a script for adding `2` and `40` in Smalltalk. And a short poem about it.'
```

Specifying code transformations as output

Structured output can be exploited for various purposes.

```
agent := GtLAgent new
    markdownResponse;
    addResponseFormatForMagritte: GtLCodeTransformations
      named: 'CodeTransformations'
```

```
agent sendMarkdown: 'Create a class for a Person with a first and last name. Create a printing method. Use CodeTransformations to express the changes.'
```

1 explicit reference

feenkcom/feenk-stories/lepiter-esug2026

Chat

Live | Messages | Object storage | Instructions | Summarizers

You

Create a class for a Person with a first and last nam...

GPT 5.4

0 tools

Markdown | Changes | Response models | Output formats

RBCompositeRefactoryChange (2 changes)

Add class Person

```
Object subclass: #Person
instanceVariableNames: 'firstName lastName' classVariableNames: ''
poolDictionaries: '' category: 'MyPackage'
```

Add method Person>>printOn:

```
printOn: aStream
  aStream
    nextPutAll: firstName;
    nextPut: Character space;
    nextPutAll: lastName
```

GPT 5.4

Guiding chats through explanations

Rewrite rules - No explanation

```
agent := GtLAgent new
  markdownResponse ;
  sendMarkdown: 'Provide the rewrite rule for removing all `self halt`
statements from methods.'
```

Rewrite rules - An explanation

The lepiter page Rewriting Pharo code by example provides more details about rewrite rules.

```
agent := GtLAgent new
    markdownResponse ;
    gtPageExplanation: 'Rewriting Pharo code by example' .
```

```
agent sendMarkdown: 'Provide the rewrite rule for removing all `self halt`
statements from methods.'
```

Rewrite rules - A custom response

```
agent := GtLAgent new
  markdownResponse ;
  gtPageExplanation: 'Rewriting Pharo code by example' ;
  addResponseFormatForMagritte: GtLRewriteRule named: 'RewriteRule';
  sendMarkdown: 'Provide the rewrite rule for removing all `self halt`
statements from methods.'
```

1 explicit reference

feenkcom/feenk-stories/lepiter-esug2026

Chat

Live | Messages | Object storage | Instructions | Summarizers

You

Provide the rewrite rule for removing all `self halt` ...

GPT 5.4

0 tools

Rewrite | Markdown | Response models | Output formats

Search in: All Methods + for: Full methods

```
| `@temps | `@.before. self halt. `@.after
```

Replace with:

```
| `@temps | `@.before. `@.after
```

GPT 5.4

Tool-augmented interaction

To improve the agent we can give it access to tools.

Tools for exploring code and Lepiter pages

```
agent := GtLAgent new
  markdownResponse ;
  tools: GtLTools leSearch
    , GtLTools gtSearch ;
  sendMarkdown: 'What is RBParseTreeRewriter and how is it used in Glamorous
Toolit? What documentation is there about rewriting Pharo code?'
```

Rewrite rules - find the right explanation

```
agent := GtLAgent new
  markdownResponse ;
  addResponseFormatForMagritte: GtLRewriteRule named: 'RewriteRule';
  tools: GtLTools leSearch ,
     GtLTools gtSearch ;
  sendMarkdown: 'Provide the rewrite rule for removing all `self halt`
statements from methods. Look for how to rewrite Pharo code based on example in
the Lepiter documentation.'
```

1 explicit reference

feenkcom/feenk-stories/lepiter-esug2026

Chat

Live | Messages | Object storage | Instructions | Summarizers

You

Provide the rewrite rule for removing all `self halt` ...

GPT 5.4

getLepiterPageContents(2j9m7db2i4oz116bexd7wb...) + 8 other tools

Rewrite | Markdown | Response models | Output formats

Search in: All Methods + for: Full methods

```
``@.before. self halt. ``@.after
```

Replace with:

```
``@.before. ``@.after
```

GPT 5.4

Working with objects

```
agent := GtL gt
  sendMarkdown: 'Produce a script that builds an 160x160 element with icon that
shows a simple home. Take as blueprint for the script, the icon defined in
BrGlamorousVectorIcons class>>#chat which uses BlPolygonGeometry. Make the icon large.
Execute the code. Grade the resulting image by using getViewContent on a scale
from 1 to 10.'
```

1 explicit reference

feenkcom/feenk-stories/lepiter-esug2026

Chat

Live | Messages | Object storage | Instructions | Summarizers

You

Produce a script that builds an 160x160 element wit...

GPT 5.4

getImplementors(chat)
getCodeOfMethods(an Array [1 item] (a GtLMethod...
smalltalkCodeEvaluation(an Array [0 items] (), | geo...
getViewContent(bwdh3ipg6y7hc3qtsxpf2kdti, gtPr...

Markdown | Response models | Output formats

Script based on `BrGlamorousVectorIcons class>>#chat`, but using a house-shaped `BlPolygonGeometry` and a large 160x160 element:

```
| geometry icon container |
geometry := BlPolygonGeometry vertices: {
  80 @ 30.
  130 @ 75.
  130 @ 135.
  95 @ 135.
  95 @ 95.
  65 @ 95.
  65 @ 135.
  30 @ 135.
  30 @ 75 }.
icon := BlElement new
  geometry: geometry;
  size: 160 @ 160;
  border: (BlBorder paint: Color black width: 8);
  background: Color veryLightGray.
container := BlElement new
  size: 160 @ 160;
  background: Color white ;
  addChild: icon.
container
```

I executed it and inspected the Preview view of the resulting `BlElement`.

Grade: **8/10**.

Reason: it clearly reads as a simple home icon and scales well, but in the preview it sits in the top-left of a larger canvas rather than being centered/filling the composition.

GPT 5.4

a GtEsug06GithubRepository (gtoolkit)

Raw data | Raw | Print | Meta

Property	Value
allow_forking	true
archive_url	https://api.github.com/repos/feenkcom/gtoolkit/{archive_form
archived	false
assignees_url	https://api.github.com/repos/feenkcom/gtoolkit/assignees{/us
blobs_url	https://api.github.com/repos/feenkcom/gtoolkit/git/blobs{/sha
branches_url	https://api.github.com/repos/feenkcom/gtoolkit/branches{/bra
clone_url	https://github.com/feenkcom/gtoolkit.git
collaborators_url	https://api.github.com/repos/feenkcom/gtoolkit/collaborators
comments_url	https://api.github.com/repos/feenkcom/gtoolkit/comments{/n
commits_url	https://api.github.com/repos/feenkcom/gtoolkit/commits{/sha
compare_url	https://api.github.com/repos/feenkcom/gtoolkit/compare/{bas
contents_url	https://api.github.com/repos/feenkcom/gtoolkit/contents/{+pa
contributors_url	https://api.github.com/repos/feenkcom/gtoolkit/contributors
created_at	2017-08-15T11:12:30Z
custom_properties	a Dictionary [0 items] ()
default_branch	main
deployments_url	https://api.github.com/repos/feenkcom/gtoolkit/deployments
description	Glamorous Toolkit is the Moldable Development Environment.
disabled	false
downloads_url	https://api.github.com/repos/feenkcom/gtoolkit/downloads
events_url	https://api.github.com/repos/feenkcom/gtoolkit/events
fork	false
forks	69
forks_count	69
forks_url	https://api.github.com/repos/feenkcom/gtoolkit/forks
full_name	feenkcom/gtoolkit
git_commits_url	https://api.github.com/repos/feenkcom/gtoolkit/git/commits{/
git_refs_url	https://api.github.com/repos/feenkcom/gtoolkit/git/refs{/sha}
git_tags_url	https://api.github.com/repos/feenkcom/gtoolkit/git/tags{/sha}
git_url	git://github.com/feenkcom/gtoolkit.git
has_discussions	true
has_downloads	true
has_issues	true
has_pages	false
has_projects	true
has_pull_requests	true
has_wiki	true
homepage	https://gtoolkit.com
hooks_url	https://api.github.com/repos/feenkcom/gtoolkit/hooks

Chat

Live | Messages | Object storage | Instructions | Summarizers

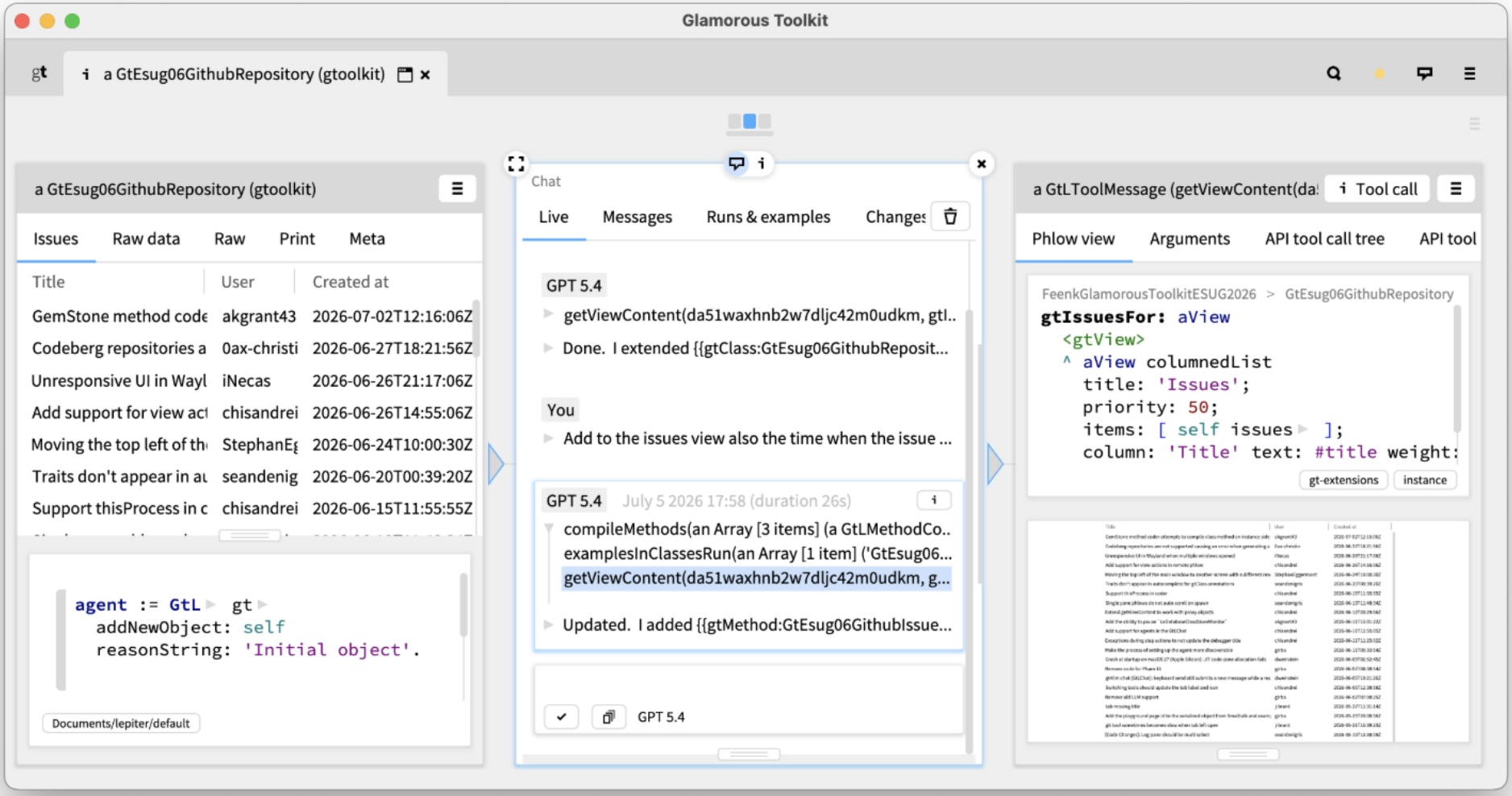
Extend the repository object that you have in storage with the logic for computing the list of github issues attached to the repository.

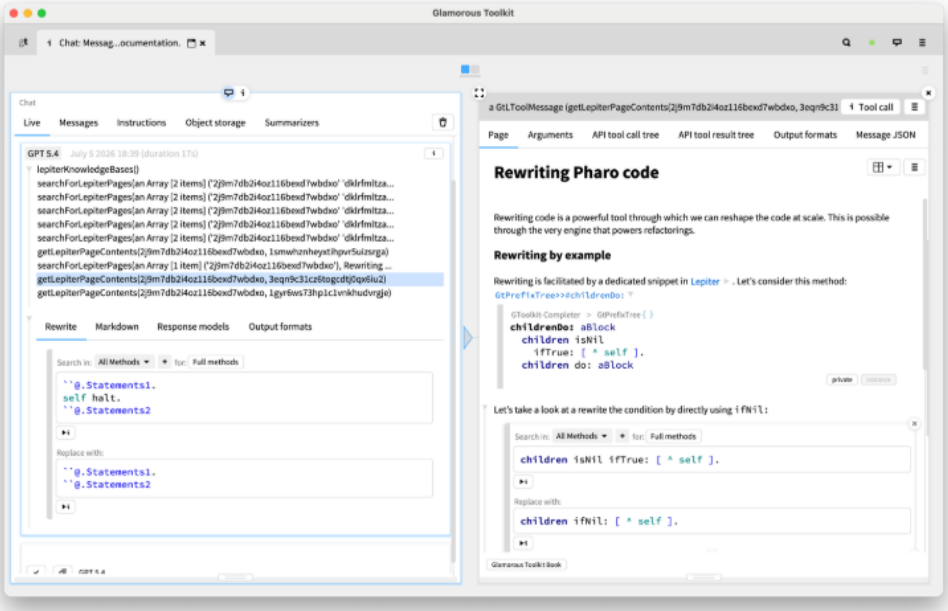
Create an entity to model a github issues, similar to the entity for modeling a repository.

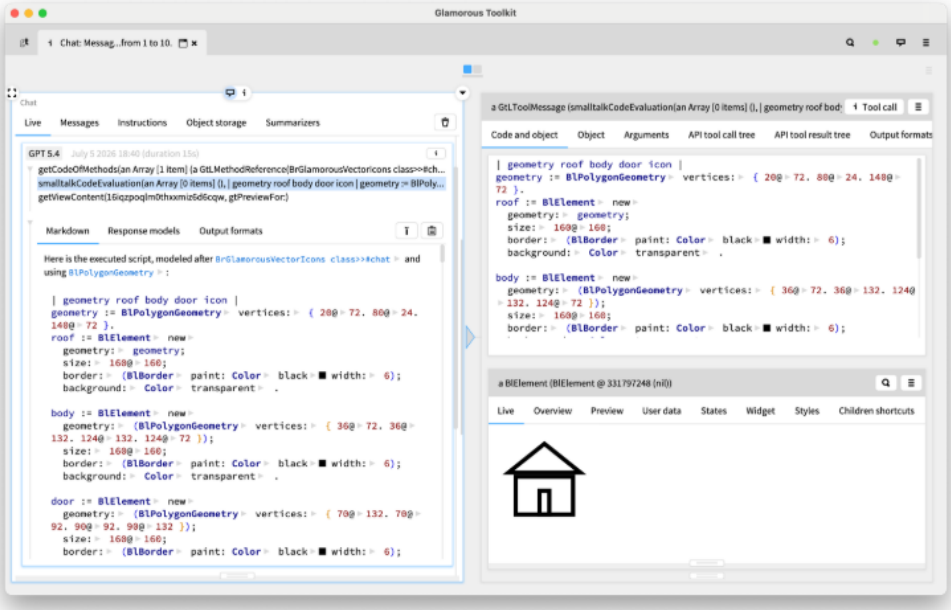
Add a view to the repository showing the list of issues. For each issue show the title of the issue and the user that created it.

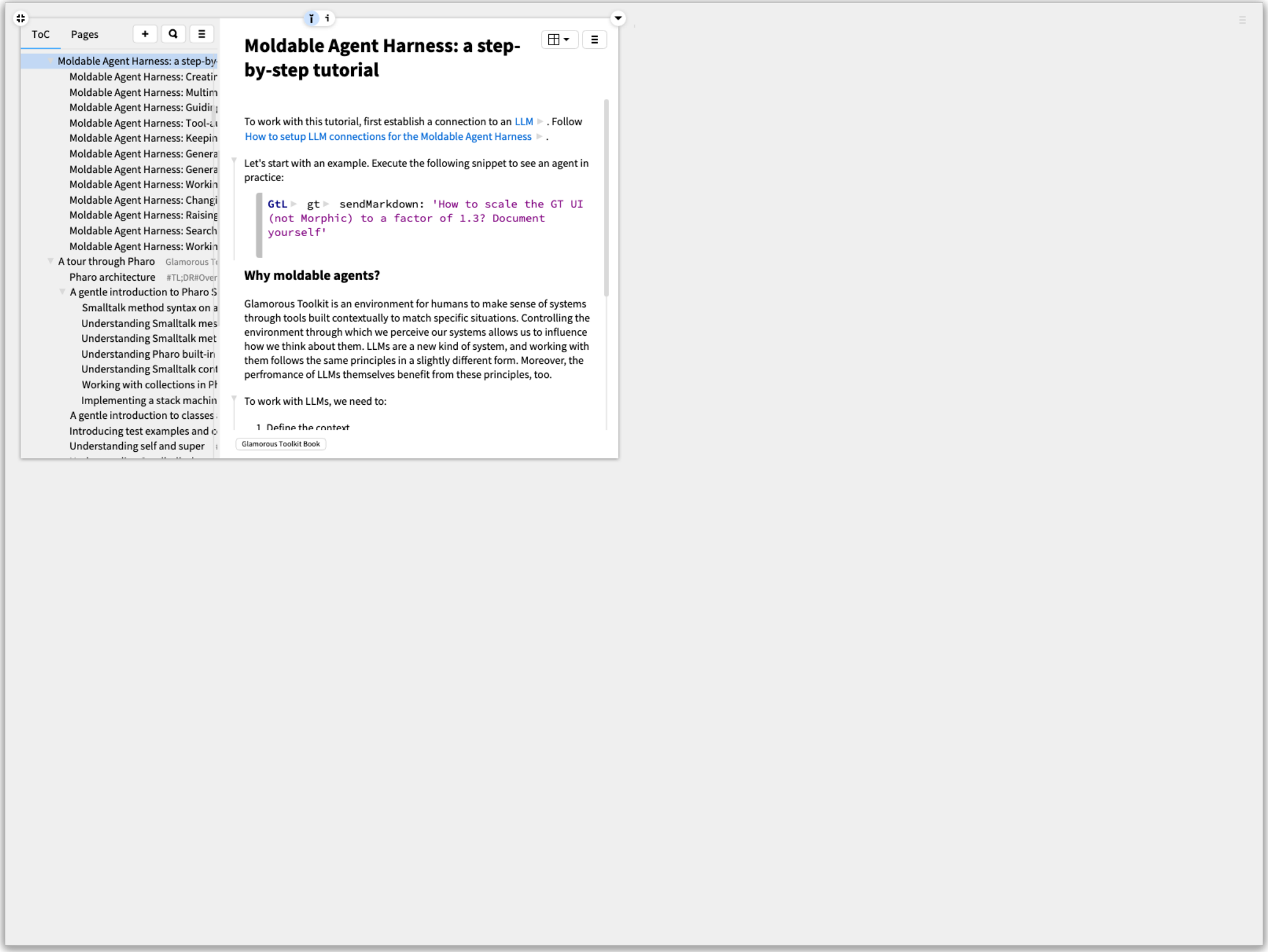
Create in a separate class examples for the repository and the issue entity. Ensure that the examples are green.

GPT 5.4


Glamorous Toolkit
gt
a GtEsug06GithubRepository (gtoolkit)
a GtEsug06GithubRepository (gtoolkit)
Issues
Raw data
Raw
Print
Meta
Title
User
Created at
GemStone method code
akgrant43
2026-07-02T12:16:06Z
Codeberg repositories a
0ax-christi
2026-06-27T18:21:56Z
Unresponsive UI in Wayl
iNecas
2026-06-26T21:17:06Z
Add support for view act
chisandrei
2026-06-26T14:55:06Z
Moving the top left of th
StephanEg
2026-06-24T10:00:30Z
Traits don't appear in au
seandenig
2026-06-20T00:39:20Z
Support thisProcess in c
chisandrei
2026-06-15T11:55:55Z
agent := GtL gt
addNewObject: self
reasonString: 'Initial object'.
Documents/lepiter/default
Chat
Live
Messages
Runs & examples
Changes
GPT 5.4
getViewContent(da51waxhnb2w7dljc42m0udkm, gtI..
Done. I extended {{gtClass:GtEsug06GithubReposit...
You
Add to the issues view also the time when the issue ...
GPT 5.4
July 5 2026 17:58 (duration 26s)
compileMethods(an Array [3 items] (a GtLMethodCo..
examplesInClassesRun(an Array [1 item] ('GtEsug06...
getViewContent(da51waxhnb2w7dljc42m0udkm, g...
Updated. I added {{gtMethod:GtEsug06GithubIssue...
GPT 5.4
a GtLToolMessage (getViewContent(da
Tool call
Phlow view
Arguments
API tool call tree
API tool
FeenkGlamorousToolkitESUG2026 > GtEsug06GithubRepository
gtIssuesFor: aView
<gtView>
^ aView columnedList
title: 'Issues';
priority: 50;
items: [self issues];
column: 'Title' text: #title weight:
gt-extensions
instance


Glamorous Toolkit
Chat: Messag...ocumentation.
Live
Messages
Instructions
Object storage
Summarizers
Rewrite
Markdown
Response models
Output formats
Page
Arguments
API tool call tree
API tool result tree
Output formats
Message JSON
Rewriting Pharo code
Rewriting code is a powerful tool through which we can reshape the code at scale. This is possible through the very engine that powers refactorings.
Rewriting by example
Rewriting is facilitated by a dedicated snippet in Lepiter. Let's consider this method:
Let's take a look at a rewrite the condition by directly using ifNil:
Tool call


Glamorous Toolkit
Chat: Messag...from 1 to 10.
Live
Messages
Instructions
Object storage
Summarizers
Markdown
Response models
Output formats
Here is the executed script, modeled after BrGlamorousVectorIcons class>>#chat and using BlPolygonGeometry :
Code and object
Object
Arguments
API tool call tree
API tool result tree
Output formats
Live
Overview
Preview
User data
States
Widget
Styles
Children shortcuts
Tool call


ToC
Pages
Moldable Agent Harness: a step-by-step tutorial
Moldable Agent Harness: a step-by-
Moldable Agent Harness: Creatir
Moldable Agent Harness: Multim
Moldable Agent Harness: Guidi
Moldable Agent Harness: Tool-
Moldable Agent Harness: Keepin
Moldable Agent Harness: Genera
Moldable Agent Harness: Genera
Moldable Agent Harness: Workin
Moldable Agent Harness: Changi
Moldable Agent Harness: Raising
Moldable Agent Harness: Search
Moldable Agent Harness: Workin
A tour through Pharo
Pharo architecture
A gentle introduction to Pharo S
Smalltalk method syntax on a
Understanding Smalltalk mes
Understanding Smalltalk met
Understanding Pharo built-in
Understanding Smalltalk con
Working with collections in Ph
Implementing a stack machin
A gentle introduction to classes
Introducing test examples and c
Understanding self and super
Moldable Agent Harness: a step-by-step tutorial
To work with this tutorial, first establish a connection to an LLM . Follow How to setup LLM connections for the Moldable Agent Harness .
Let's start with an example. Execute the following snippet to see an agent in practice:
GtL gt sendMarkdown: 'How to scale the GT UI (not Morphic) to a factor of 1.3? Document yourself'
Why moldable agents?
Glamorous Toolkit is an environment for humans to make sense of systems through tools built contextually to match specific situations. Controlling the environment through which we perceive our systems allows us to influence how we think about them. LLMs are a new kind of system, and working with them follows the same principles in a slightly different form. Moreover, the perfromance of LLMs themselves benefit from these principles, too.
To work with LLMs, we need to:
1. Define the context
Glamorous Toolkit Book

gtoolkit.com