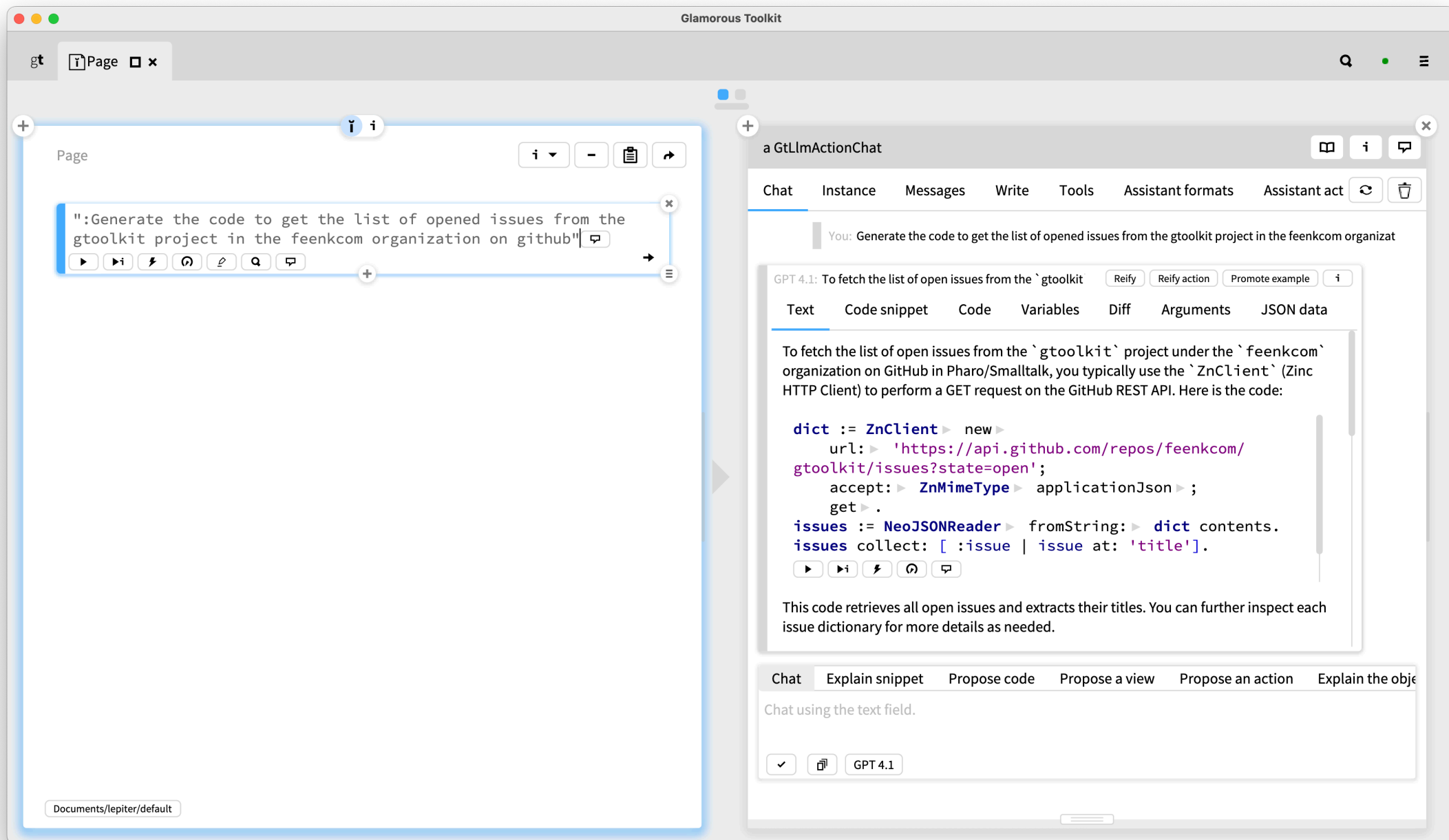


AI assistants in Glamorous Toolkit
andrei.chis@feenk.com



gt

AI assistants - presentation script

Q

•

≡

+

a GtGitHubIssues

📖

i

💬

GitHub Issues

Raw

Print

Meta

Class

GtGitHubIssues

-

🔗

📖

🔗

🔗

📄

💬

Superclass: Object

Package: GtGitHub

Tag:

Comment

References

References in snippets

Advice definitions

🔍

+

i

Q

Methods up to

GtGitHubIssues

Category

All methods

Instance slots

All methods

read/write

+

defaultOrganization

llm-generated

instance

defaultRepository

llm-generated

instance

fetchIssues

llm-generated

instance

gtIssuesListFor: aView

```
<gtView>
^ aView list
  title: 'GitHub Issues';
  items: [ issues ];
  itemText: [ :each | each at: 'title' ]
```

✓

-

i

📄

Q

💬

▶i

llm-generated

instance

initialize

llm-generated

instance

issuesTitles

llm-generated

instance

Open chat

+

a GtLlmActionChat

📖

i

💬

Chat

Instance

Messages

Write

Tools

Assistant formats

Assistant act

↺

🗑️

You: Convert this view method from a list into a columned list, and include a column showing the title and

GPT 4.1: To convert the current list view to a columned list with separate

Reify

Promote example

i

Text

Code

JSON data

Changes

▼

✕

RBCompositeRefactoryChange

✕

GtGitHubIssues>>gtIssuesListFor:

gtIssuesListFor: aView

```
<gtView>
^ aView list
  title: 'GitHub
Issues';
  items: [ issues ];
  itemText: [ :each |
each at: 'title' ]
```

gtIssuesListFor: aView

```
<gtView>
^ aView columnedList
  title: 'GitHub
Issues';
  items: [ issues ];
  column: 'Title'
  text: [ :each | each at:
'title' ];
  column: 'Author'
  text: [ :each | (each at:
'user') at: 'login' ]
```

✓

Chat

Rename method

Review method

Propose code

Explain the object

Explain the me

Chat using the text field.

✓

📄

GPT 4.1

gt

Building an ...ng blog posts

Building an LLM assistant for editing blog posts

LLMs work interestingly well for shallow questions (I call shallow the questions whose answers can be evaluated against reality quickly). Moldable Development splits technical decision-making into small, shallow problems addressed by small, shallow tools. As such, it can be a fertile target for LLMs.

`blogPostText := thisSnippet previousSibling ▶ text`

And now, let's say we want to have a chat about it with a dedicated LLM:

`GtOpenAIBlogPostAssistant ▶ new ▶ createChatOn: ▶ blogPostText`

Building a dedicated chat ... while chatting

The above assistant has been evolved from a generic assistant. We can refine a generic assistant to get to a similar result. This page demonstrates how to do that.

We start by instantiating a generic assistant with a description.

`assistant := GtLLMAssistant ▶ new ▶
description: ▶ 'You are an assistant that is used to
interactively work on blog posts.'`

We want the assistant to work with a Post format, so we create it and give information on how to serialize and deserialize the blog post for the assistant.

Glamorous Toolkit Book

a GtLLMActionChat

Chat

Instance

Messages

Write

Tools

Assistant formats

Assistant act

You: Correct grammar

GPT 4.1: Correct grammar

i

Action

Reify

Promote example

i

Post

Diff

JSON data

There are at least two ways to use LLMs to support the understanding of software systems:
(A) use LLMs to generate summaries
(B) use LLMs to generate tools with which you generate summaries

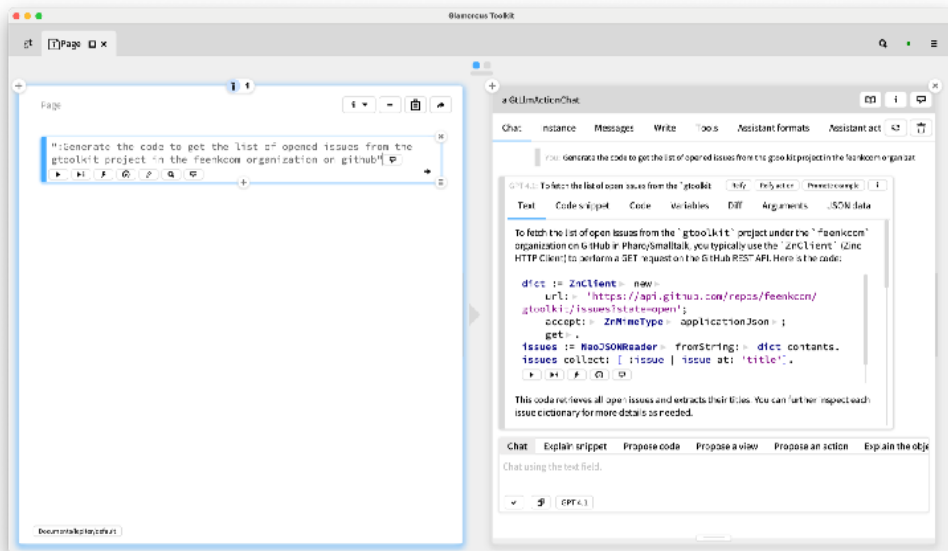
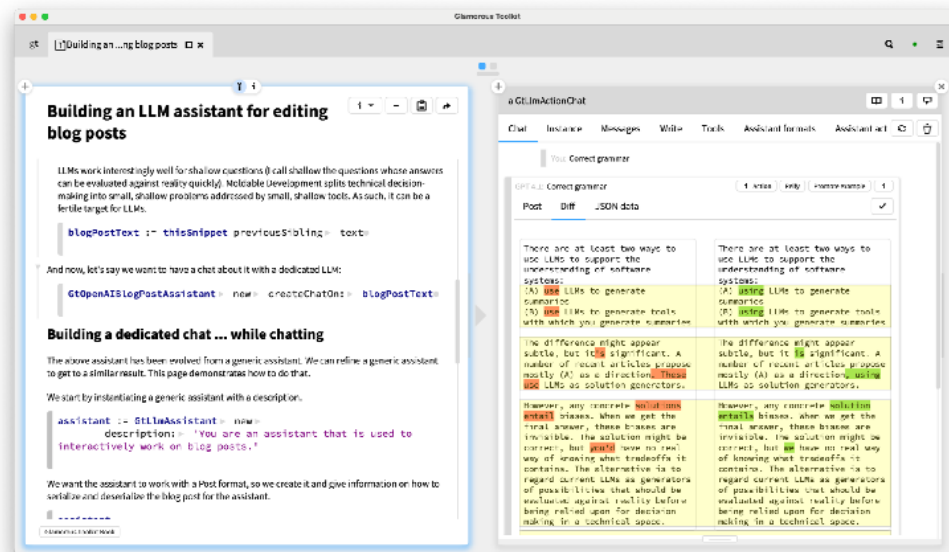
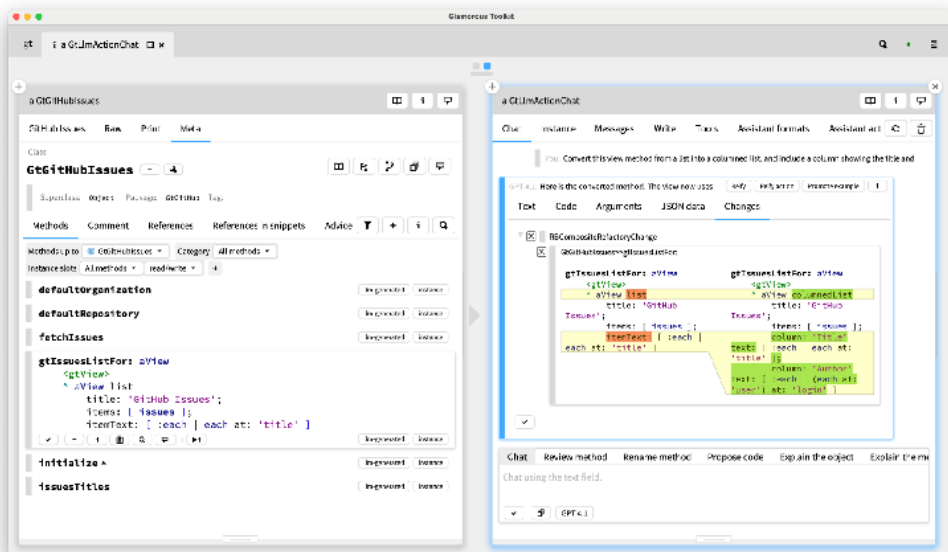
There are at least two ways to use LLMs to support the understanding of software systems:
(A) using LLMs to generate summaries
(B) using LLMs to generate tools with which you generate summaries

The difference might appear subtle, but it's significant. A number of recent articles propose mostly (A) as a direction. These use LLMs as solution generators.

The difference might appear subtle, but it is significant. A number of recent articles propose mostly (A) as a direction, using LLMs as solution generators.

However, any concrete solutions entail biases. When we get the final answer, these biases are invisible. The solution might be correct, but you'd have no real way of knowing what tradeoffs it contains. The alternative is to regard current LLMs as generators of possibilities that should be evaluated against reality before being relied upon for decision-making in a technical space.

However, any concrete solution entails biases. When we get the final answer, these biases are invisible. The solution might be correct, but we have no real way of knowing what tradeoffs it contains. The alternative is to regard current LLMs as generators of possibilities that should be evaluated against reality before being relied upon for decision-making in a technical space.



gtoolkit.com