

哈尔滨工业大学 计算学部

2024 年秋季学期 《开源软件开发实践》

Lab3： 自选开源项目编程

学号	姓名	联系方式
2022210939	岳翼博	19716318612
2022211709	曾湘临	18588602616
2022211473	申亚轩	19819824901
2022211690	荣文杰	18872333356

目 录

1 实验要求.....	1
2 实验内容 1 掌握基本 Markdown 语法	1
3 实验内容 2 参与开源项目	8
3.1 分析所选项目	8
3.2 项目参与者贡献过程分析	12
3.3 贡献代码	14
4 小结	18

1 实验要求

实验手册中要求达到的实验目标与过程如下：

实验目标：

1. 掌握基本的 Markdown 语法。
2. 能够根据开源项目提供的文档，对项目进行了解和分析。
3. 实际参与某开源项目，贡献代码。

实验过程：

1. 掌握基本 Markdown 语法：
 - 1) 阅读学习提供的“简单教程”或其他文档，了解 Markdown 的基本语法。
 - 2) 通过“在线练习”网站，练习并掌握基本 Markdown 的使用，完成所有的练习任务。
2. 参与开源项目：
 - 1) 寻找开源项目：选择 star 数 20+ 的开源项目，或从教师推荐的参考开源项目列表中选择。
 - 2) 参与开源项目开发：选择某开源项目的 issue 进行参与，issue 必须同修改代码相关，或自行提出 issue 并提交 PR。
 - 3) 分析项目：从项目文档和 GitHub 分析功能，分析项目结构、参与贡献的要求、项目协议、项目活跃度、贡献者多少等。
 - 4) 分析参与者：选择项目某个参与者，分析其提交 PR 的过程。
 - 5) 贡献代码：选择项目的某个 issue 或自行提出 issue，针对选择的 issue，通过 PR 的方式对项目贡献代码。

2 实验内容 1 掌握基本 Markdown 语法

1. 斜体和粗体

- (1) 斜体：要在 Markdown 中将一个短语变成斜体，可以在单词周围加上下划线 (_)。例如， `_this_` 单词将变成斜体。

For this next lesson, make the word "not" italic.

Writing in Markdown is not that hard!

Writing in Markdown is *not* that hard!

Skip

- (2) 粗体：要在 Markdown 中将短语加粗，可以在单词周围加上两个星号 (**)。这将 `**really**` 表达你的意思

(3) 混用：同一行：

In the box below, make the words "Of course" italic, and the words "a little moxie" bold.

```
"_Of course_" she whispered. Then, she shouted: "All I need  
is **a little moxie**!"
```

"*Of course*," she whispered. Then, she shouted: "All I need is
a little moxie!"

Skip

同一句：

In general, it doesn't matter which order you place the asterisks or underscores. In the box below, make the words "This is unbelievable" both bold and italic. Place the asterisks ****_on the outside_****, just to make it more legible.

```
If you're thinking to yourself, **_This is unbelievable_**,  
you'd probably be right.
```

If you're thinking to yourself, ***This is unbelievable***, you'd
probably be right.

Skip

Now you know how to make things bold and italic in Markdown!

On to the next lesson!

2. 标题

- (1) 标题：在 Markdown 中制作标题时，需要在短语前加上散列标记（#）。散列标记的数量与所需标题的大小相同。例如，标题一用一个散列号（# Header One），标题三用三个散列号（### Header Three）

For this next lesson, make each header the right size.

```
# Header one  
## Header two  
### Header three  
#### Header four  
##### Header five  
##### Header six
```

Header one

Header two

Header three

Header four

Header five

Header six

Skip

- (2) 混用：在标题中混合使用内联样式，例如斜体。在下面的方框中，将第一行设为四级标题，并用斜体标注书名

You can also mix and match inline styles within headers, such as *italicizing* them. In the box below, make the first line a heading level four, and italicize the name of the book:

<pre>#### Colombian Symbolism in <u>One Hundred Years of Solitude</u> Here's some words about the book <u>One Hundred Years...</u></pre>	Colombian Symbolism in <i>One Hundred Years of Solitude</i> Here's some words about the book <i>One Hundred Years...</i>
--	---

Skip

And that's all there is to making headers in Markdown.

On to the next lesson!

3. 链接

- (1) 内联链接: 创建内联链接, 先用括号 ([]) 将链接文本包起来, 然后用括号 (()) 将链接包起来

In the box below, make a link to www.google.com, with link text that says "Search for it."

<pre>[Search for it.](www.google.com)</pre>	Search for it.
---	--

Skip

- (2) 加强调:

You can add emphasis to link texts, if you like. In the box below, make the phrase "really, really" bold, and have the entire sentence link to www.dailykitten.com. You'll want to make sure that the bold phrasing occurs within the link text brackets.

<pre>[You're really, really going to want to see this.](www.dailykitten.com)</pre>	You're really, really going to want to see this.
---	--

Skip

- (3) 标题中创建:

Fantastic!

Although it might make for an awkward experience, you can make links within headings, too.

For this next tutorial, make the text a heading four, and turn the phrase "the BBC" into a link to www.bbc.com/news:

```
#### The Latest News from [the BBC](www.bbc.com/news)
```

The Latest News from [the BBC](http://www.bbc.com/news)

Skip

(4) 引用链接：这种链接实际上是对文档中另一处内容的引用

In the box below, we've started writing out some reference links. You'll need to finish them up! Call the first reference tag "a fun place", and make it link to www.zombo.com; make the second link out to www.stumbleupon.com.

```
Do you want to [see something fun][a fun place]?  
Well, do I have [the website for you][another fun place]!  
[a fun place]: www.zombo.com  
[another fun place]: www.stumbleupon.com
```

Do you want to [see something fun](#)?

Well, do I have [the website for you](#)!

Skip

You now know how to make links in Markdown!

On to the next lesson!

4. 图片

- (1) 内联图像链接：要创建内嵌图片链接，请输入感叹号（!），用括号（[]）封装 alt 文本，然后用括号（()）封装链接。（Alt 文本是为视障人士描述图像的短语或句子）

In the box below, turn the link to an image, and fill out the alt text brackets to say "A pretty tiger":

```
![[A pretty tiger]](https://upload.wikimedia.org/wikipedia/commons/5/56/Tiger.50.jpg)
```



Skip

- (2) 引用链接：

In the box below, we've started placing some reference images; you'll need to complete them, just like the last lesson. Call the first reference tag "Black", and make it link to https://upload.wikimedia.org/wikipedia/commons/a/a3/81_INF_DIV_SSI.jpg; make the second image link out to <http://icons.iconarchive.com/icons/google/noto-emoji-animals-nature/256/22221-cat-icon.png>.

```
![[Black cat]][Black]

![[Orange cat]][Orange]

[Black]: https://upload.wikimedia.org/wikipedia/commons/a/a3/81_INF_DIV_SSI.jpg
[Orange]: http://icons.iconarchive.com/icons/google/noto-emoji-animals-nature/256/22221-cat-icon.png
```



[Skip](#)

Ta da! You've learned all there is to adding images in Markdown!

[On to the next lesson!](#)

5. 引用

- (1) 正文引用：要创建正文引用，只需在行前加上 "大于 "符号（>）即可

In the box below, turn the book quotation into a blockquote:

```
I read this interesting quote the other day:

>"Her eyes had called him and his soul had leaped at the
call. To live, to err, to fall, to triumph, to recreate life
out of life!"
```

```
I read this interesting quote the other day:

    "Her eyes had called him and his soul had
    leaped at the call. To live, to err, to fall, to
    triumph, to recreate life out of life!"
```

[Skip](#)

- (2) 跨段落引用：在引文的每一行放置一个字符

In the box below, Make the entire quotation a block quote by inserting a caret on each line.

```
>Once upon a time and a very good time it was there was a
moocow coming down along the road and this moocow that was
coming down along the road met a nicens little boy named baby
tuckoo...
>
>His father told him that story: his father looked at him
through a glass: he had a hairy face.
>
>He was baby tuckoo. The moocow came down the road where
Betty Byrne lived: she sold lemon platt.
```

Once upon a time and a very good time it was there was a moocow coming down along the road and this moocow that was coming down along the road met a nicens little boy named baby tuckoo...

His father told him that story: his father looked at him through a glass: he had a hairy face.

He was baby tuckoo. The moocow came down the road where Betty Byrne lived: she sold lemon platt.

Skip

(3) 引文可以包含其他 Markdown 元素，如斜体、图像或链接

In the box below, make the French text italic (not including the exclamation point). Also, turn the entire quote into a blockquote.

```
>He left her quickly, fearing that her intimacy might turn to
jibing and wishing to be out of the way before she offered
her ware to another, a tourist from England or a student of
Trinity. Grafton Street, along which he walked, prolonged
that moment of discouraged poverty. In the roadway at the
head of the street a slab was set to the memory of Wolfe Tone
and he remembered having been present with his father at its
laying. He remembered with bitterness that scene of tawdry
tribute. There were four French delegates in a brake and one,
a plump smiling young man, held, wedged on a stick, a card on
which were printed the words: _VIVE L'IRLANDE_!
```

He left her quickly, fearing that her intimacy might turn to jibing and wishing to be out of the way before she offered her ware to another, a tourist from England or a student of Trinity. Grafton Street, along which he walked, prolonged that moment of discouraged poverty. In the roadway at the head of the street a slab was set to the memory of Wolfe Tone and he remembered having been present with his father at its laying. He remembered with bitterness that scene of tawdry tribute. There were four French delegates in a brake and one, a plump smiling young man, held, wedged on a stick, a card on which were printed the words: *VIVE L'IRLANDE!*

Skip

Ta da! You've learned all there is to creating blockquotes in Markdown!

On to the next lesson!

6. 列表

(1) 无序列表：需要在列表中的每个项目前加上星号（*）

In the box below, turn the words separated by a comma into a list.

```
* Flour
* Cheese
* Tomatoes
```

- Flour
- Cheese
- Tomatoes

Skip

(2) 有序列表：有序列表的前缀是数字，而不是星号

In the box below, turn the rest of the recipe into an ordered list.

```
1. Cut the cheese
2. Slice the tomatoes
3. Rub the tomatoes in flour
```

```
1. Cut the cheese
2. Slice the tomatoes
3. Rub the tomatoes in flour
```

Skip

(3) 混用：在列表中添加斜体、粗体或链接

You can choose to add italics, bold, or links within lists, as you might expect. In the box below, turn the Latin names for the plants into italics.

```
* Azalea (_Ericaceae Rhododendron_)
* Chrysanthemum (_Anthemideae Chrysanthemum_)
* Dahlia (_Coreopsideae Dahlia_)
```

```
• Azalea (Ericaceae Rhododendron)
• Chrysanthemum (Anthemideae Chrysanthemum)
• Dahlia (Coreopsideae Dahlia)
```

Skip

(4) 嵌套列表：每个星号都要比前一项缩进一个空格

In the box below, turn the character's characteristics into sub-bullets.

```
* Calculus
  * A professor
  * Has no hair
  * Often wears green
* Castafiore
  * An opera singer
  * Has white hair
  * Is possibly mentally unwell
```

```
• Calculus
  • A professor
  • Has no hair
  • Often wears green
• Castafiore
  • An opera singer
  • Has white hair
  • Is possibly mentally unwell
```

Skip

(5) 列表和缩进：假设想创建一个弹出式列表，它需要一些额外的上下文（但不是另一个列表）

In the box below, convert the bullet points into their own paragraphs.

```
1. Cut the cheese
   Make sure that the cheese is cut into little triangles.

2. Slice the tomatoes
   Be careful when holding the knife.

   For more help on tomato slicing, see Thomas Jefferson's
   seminal essay Tom Ate Those.
```

```
1. Cut the cheese
   Make sure that the cheese is cut into little triangles.

2. Slice the tomatoes
   Be careful when holding the knife.

   For more help on tomato slicing, see Thomas
   Jefferson's seminal essay Tom Ate Those.
```

Skip

You now know how to make lists in Markdown!

On to the next lesson!

7. 段落

(1) 软中断：在每行之后插入两个空格来实现这一目的

Let's try this technique out. In the box below, insert the necessary number of spaces to make the poem render correctly:

```
We pictured the meek mild creatures where  
They dwelt in their strawy pen,  
Nor did it occur to one of us there  
To doubt they were kneeling then. |
```

```
? We pictured the meek mild creatures where  
They dwelt in their strawy pen,  
Nor did it occur to one of us there  
To doubt they were kneeling then.
```

Skip

(2) 列表中设置：除诗歌格式外，这些软中断的常见用途之一是在列表中设置段落格式

In the box below, instead of using hard breaks, tighten the sub-paragraphs with soft breaks:

```
1. Crack three eggs over a bowl.  
Now, you're going to want to crack the eggs in such a way  
that you don't make a mess.  
If you _do_ make a mess, use a towel to clean it up!
```

```
2. Pour a gallon of milk into the bowl. |  
Basically, take the same guidance as above: don't be messy,  
but if you are, clean it up!
```

```
1. Crack three eggs over a bowl.  
Now, you're going to want to crack the eggs in such a  
way that you don't make a mess.  
If you do make a mess, use a towel to clean it up!
```

```
2. Pour a gallon of milk into the bowl.  
Basically, take the same guidance as above: don't be  
messy, but if you are, clean it up!
```

Skip

Et voila! You now know how to make soft breaks in Markdown!

On to the next lesson!

3 实验内容 2 参与开源项目

3.1 分析所选项目

项目地址: <https://github.com/wger-project/flutter>

3.1.1 开发目标

wger 是一个开源的个人健身/运动跟踪器和健身管理软件。该 Flutter 项目是 wger 的移动应用程序版本,旨在提供跨平台的移动客户端,让用户可以在移动设备上方便地使用 wger 的功能。

主要目标包括:

- 提供跨平台(iOS/Android)的移动应用
- 实现健身计划、营养跟踪、体重记录等核心功能
- 与 wger 服务器同步数据
- 提供友好的用户界面和流畅的用户体验

3.1.2. 项目结构

项目采用典型的 Flutter 应用结构:

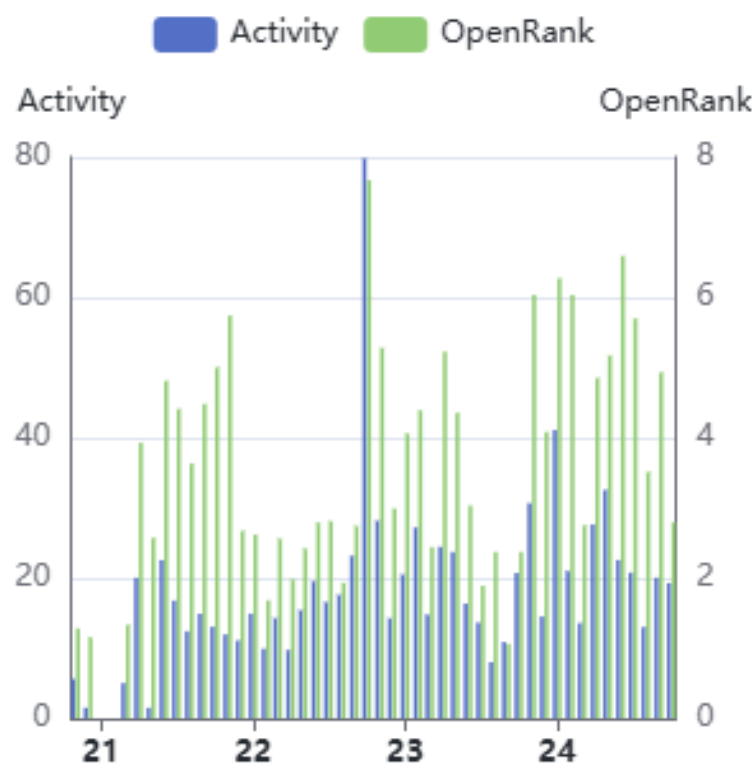
```
lib/  
|- models/      # 数据模型  
|- providers/   # 状态管理  
|- screens/     # 界面  
|- widgets/    # 可复用组件  
|- main.dart    # 入口文件
```

使用 Provider 进行状态管理,采用 MVVM 架构模式。

3.1.3. 开源协议

项目采用 AGPL-3.0 License,这是一个强 copyleft 许可证,要求修改后的软件也必须以相同许可证开源。

Activity & OpenRank Trends

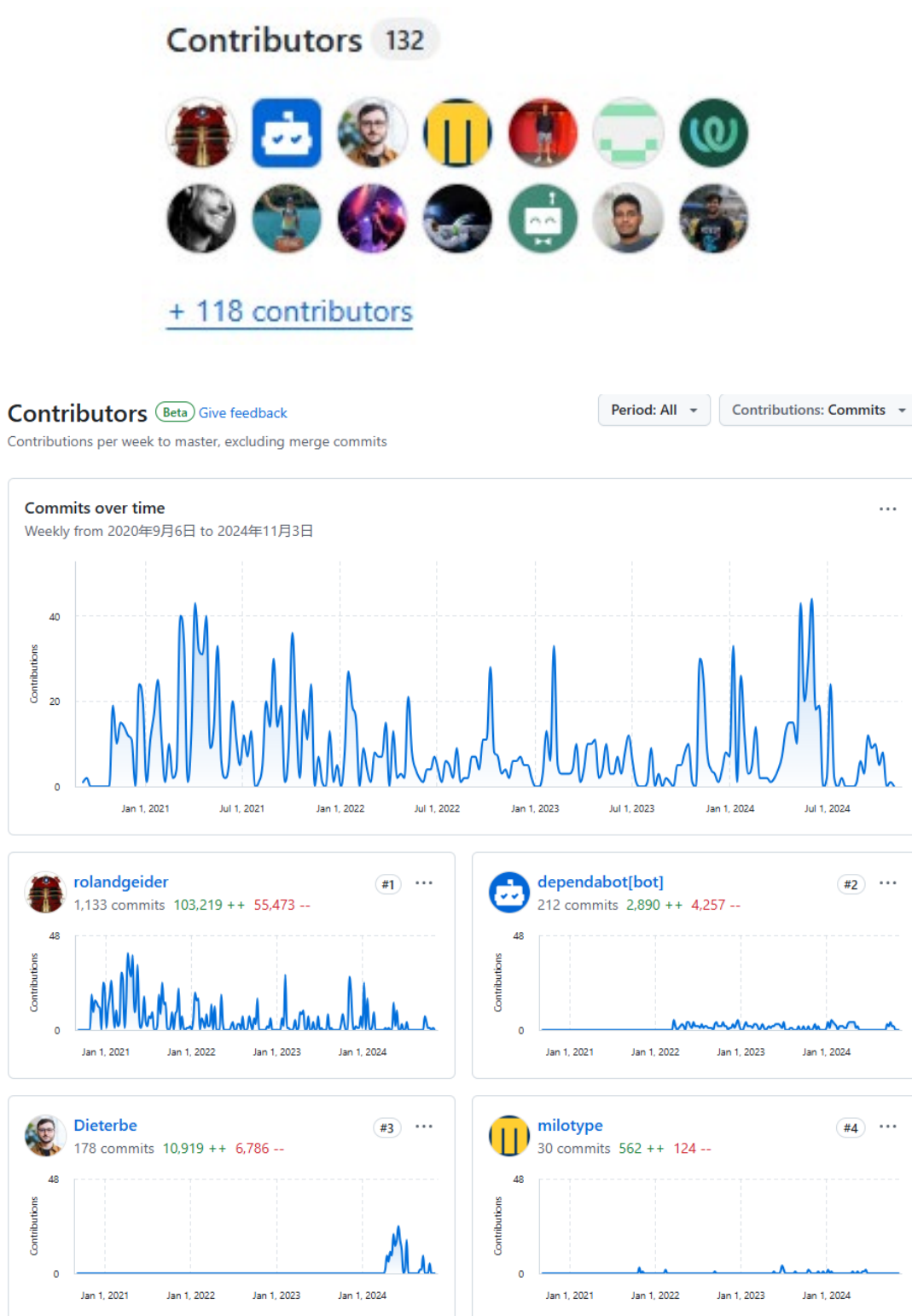


3.1.4. 项目活跃度

从活动图可以看出:

- 项目持续活跃,有稳定的提交
- 2022 年下半年活跃度明显提升
- 最近一年保持较高活跃度

3.1.5. 贡献者情况



- 总贡献者数:132
- 核心贡献者约 5-6 人
- 主要贡献者 rolandgeider 贡献了大部分代码

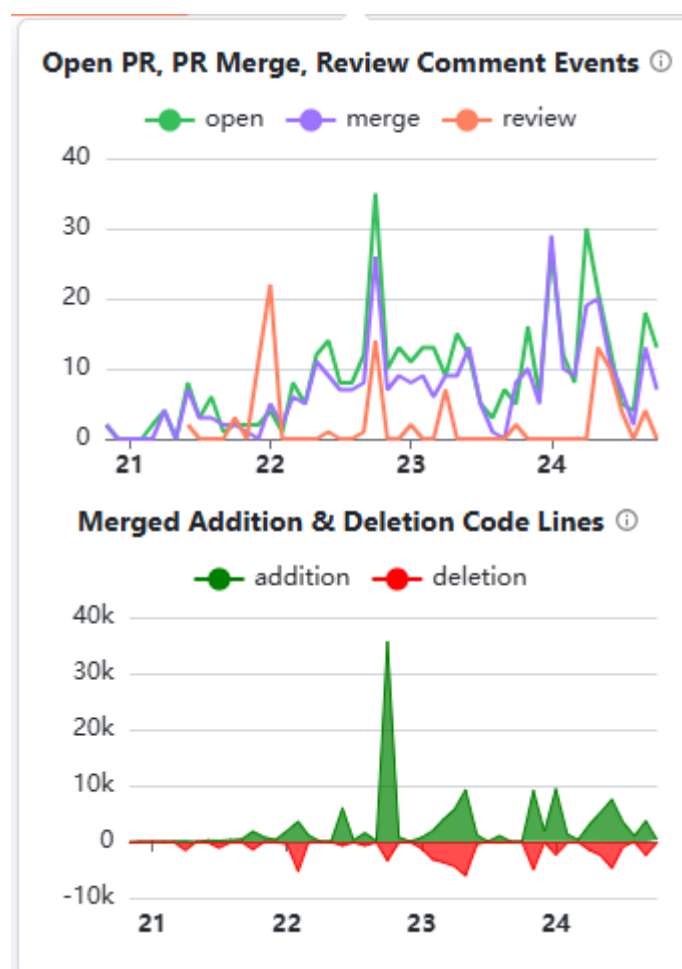
Releases 23

1.7.7 Latest
on Aug 1

+ 22 releases

3.1.6. 版本发布

- 项目有规律地发布新版本
- 最新版本 1.7.7 发布于 2024 年 8 月
- 平均每 1-2 个月发布一个新版本



3.1.7. 问题和 PR

- 开放的 Issues: 74
- 已关闭的 Issues: 125
- 开放的 Pull Requests: 19
- 已合并的 Pull Requests: 442

问题解决率和 PR 合并率都很高,说明项目维护得当。

3.1.8. 总结:

wger Flutter 项目是一个活跃的开源项目,有稳定的贡献者团队和版本发布节奏。项目结构清晰,采用现代 Flutter 开发实践。虽然贡献者总数不多,但核心团队保持着项目的持续发展。AGPL 协议可能会限制商业使用,但有利于保持项目的开源性质。总的来说,这是一个健康发展的开源项目。

3.2 项目参与者贡献过程分析

3.2.1 参与贡献的要求

要参与贡献到 wger Workout Manager 这个项目, 需要遵循以下步骤和要求:

1. 加入社区:
通过加入 Discord 服务器来与其他贡献者交流, 并了解与参与项目。
2. 安装 wger 服务器:
可以安装 wger 服务器, 最简单的方式是使用 Docker Compose。项目提供了一个链接到 Docker 项目的 GitHub 仓库。或者, 可以使用测试服务器(数据库每天重置)。
3. 安装 Flutter:
 - 1) 需要安装 Flutter 及其所有依赖项, 并创建一个新的虚拟设备。项目指定了 Flutter 的版本为 3.22。
 - 3) 访问 Flutter 官方文档来了解如何安装。
4. 设置 API 密钥:
 - 1) 如果想通过应用程序直接注册, 则需要后端设置一个用户, 该用户拥有权限。
 - 2) 创建/注册一个新用户, 生成一个 API 密钥, 并运行特定的命令来添加用户。
 - 3) 创建一个 `wger.properties` 文件并添加 API 密钥, 或者将密钥作为环境变量设置。
5. 启动应用程序:
使用 `flutter run` 命令启动应用程序, 或者使用自己的 IDE。
注意: 根据运行模拟器的方式的不同, 可能需要更改服务器的 IP 地址。
6. 运行测试:
可以使用 `flutter test` 命令来运行测试。
7. 翻译:
可以通过 Weblate 平台将应用程序翻译成你的语言。
8. 许可证:
应用程序在 GNU Affero 通用公共许可证 3 或更高版本 (AGPL 3+) 下授权, 并且有一个应用商店例外。

这些步骤概述了如何开始为 wger Workout Manager 项目做出贡献。然后对代码风格规范以及 Issue 和 Pull Request 规范没有进行要求。

3.2.2 Issue 交流和 PR 过程

Issue 地址: <https://github.com/wger-project/flutter/issues/381>

1. Issue 交流过程分析

问题提出

- 提出者: @tejasbirsingh
- 描述: @tejasbirsingh 提出了一项关于改进用户个人资料页面 UI 的功能请求,

认为当前页面的布局和设计欠缺美观。主要改进建议包括：

- 将文本/字段显示调整至页面顶部，使布局更清晰。
- 为列表视图组件添加图标，增强视觉吸引力。

方案讨论

- 参与者：@tejasbirsingh 和 @rolandgeider
- 讨论内容：
 - @tejasbirsingh 表示希望自己参与实现该功能，并描述了将字段顶部对齐及添加前导图标的具体改进方法。
 - @rolandgeider 提出了设计改进建议，建议字段位置保持在底部以便于用户拇指操作。同时，认可了图标的加入，并鼓励 @tejasbirsingh 提交 PR。
 - @tejasbirsingh 提到，考虑将电子邮件地址和验证状态合并显示，以优化空间并增加可读性，同时仍将字段保持在底部布局。

设计迭代

- 进一步设计建议：
 - @rolandgeider 提出，为避免用户误解图标的点击用途（发送验证邮件），建议在未验证状态下为该字段添加按钮，简化用户体验。
 - @tejasbirsingh 提供了两种 UI 模拟方案供参考，并进一步征求意见。
 - @rolandgeider 选择了轮廓按钮设计方案，并指出其优点在于能够区分该按钮与“保存”按钮的优先级。

最终确认

- 设计确认：@rolandgeider 最终确认了设计，并在 6 月 5 日关闭了该 Issue。

2. PR 过程分析

代码提交

- 提交者：@tejasbirsingh
- 更改内容：提交了包含图标添加、电子邮件与验证状态合并的 UI 改进。
- 关联 Issue：Fixes #381

测试验证

- 测试情况：@tejasbirsingh 本地测试通过，所有测试均成功，附上了未验证和已验证电子邮件状态的截图，确保更改符合预期。

代码审查

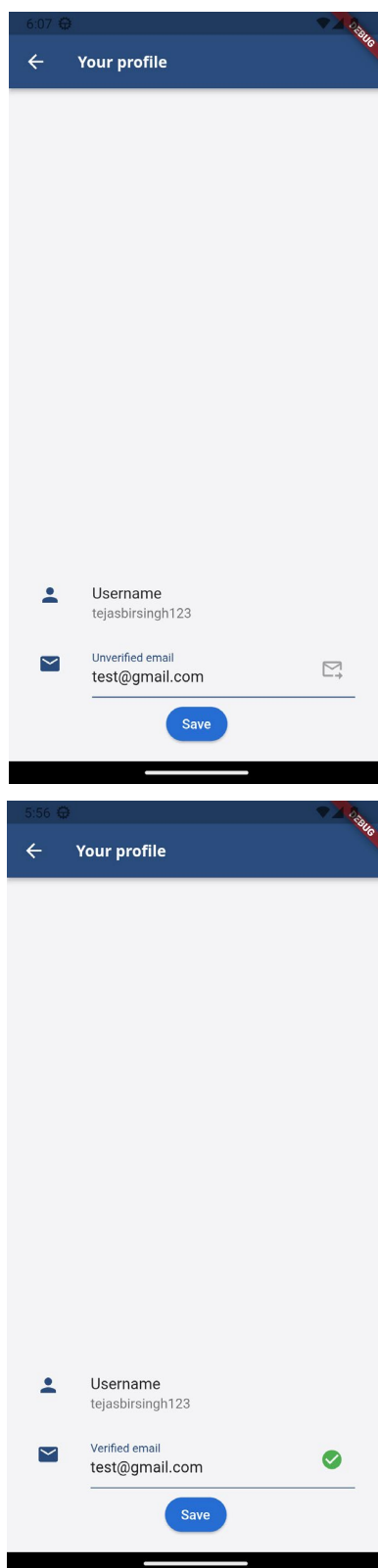
- 审查者：@rolandgeider
- 审查反馈：指出有一个测试因运行时间问题偶尔不稳定，但对代码本身无影响，因此认可代码。

PR 合并

- 合并时间：2023 年 6 月 5 日
- 完成情况：@rolandgeider 将该 PR 合并至主分支，完成了用户个人资料页面的 UI 改进。

以下两图是

未验证电子邮件状态的截图(上面)和已验证电子邮件状态的截图(下面)。



3.3 贡献代码

给出选择的 issue 或新建 issue 的地址，包含 issue 内容的截图：

选择的 issue: <https://github.com/wger-project/flutter/issues/284>

wger-project / flutter

Type to search

<> Code

Issues 74

Pull requests 20

Discussions

Actions

Projects 1

Security

Insights

Some texts are hard-coded, it is necessary to implement localization #284 New issue

Open

sandimilohanic opened on Oct 30, 2022

Use case

- images notes (file stepImages.dart)
- videos button (file video.dart)

... and other similar places.

Contribute an exercise

Variations

Description

Translation

Images

Images must be compatible with the iOS 13.0+ format. If an image is not compatible, you'll see a warning.

Only JPEG, PNG and WEBP files below 22 KB are supported.

Previous Save

Assignees

SamiraHuber

Labels

good first issue

Type

No type

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development




No branches or pull requests

Notifications

Unsubscribe

You're receiving notifications because you're subscribed to this thread.

Participants



sandimilohanic on Oct 30, 2022

Contributor Author

Hi @rolandgeider if you approve this, assign it to me. thx

rolandgeider on Oct 31, 2022

Member

good catch, we definitely want those translated as well. You can use the [same strings used in the web application](#), then weblate can suggest the same translation (even if their UI sometimes hides these suggestions a bit too much)

rolandgeider assigned sandimilohanic on Oct 31, 2022

rolandgeider added good first issue on Oct 15, 2023

SamiraHuber on Dec 28, 2023

Nothing happened here. Is it still open? I would love to do my first open source contribution and if it is still open I think this would be a good first issue for me. @rolandgeider

rolandgeider on Dec 28, 2023

Member

Hi! While it's possible that that specific string has been fixed, there might definitely be others. If you want to give it a try, go ahead

rolandgeider assigned SamiraHuber and unassigned sandimilohanic on Dec 28, 2023

FutureYL3 3 hours ago

edited by FutureYL3 · Edits

Hi @rolandgeider, is this issue still open? I would like to make my contribution to this project on this issue.

FutureYL3 mentioned this 6 minutes ago










Fixing some hard-coded texts #669

FutureYL3 1 minute ago

Hello @rolandgeider, I have submitted a PR to address some hard-coded texts I noticed while reviewing the app. Could you please take a look at it when it is convenient for you?

Add a comment

Write Preview


H B I         

Use Markdown to format your comment

Paste, drop, or click to add files

Close issue Comment

© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information



给出提交的 PR 地址，以及截图：

提交的 PR 地址：<https://github.com/wger-project/flutter/pull/669>

The screenshot shows a GitHub Pull Request (PR) #669 titled "Fixing some hard-coded texts #669" in the repository `wger-project/flutter`. The PR is open and shows a description of proposed changes, a link to issue #284, tests, and a checklist. It also displays commit history and workflow status.

Description (Proposed Changes)

- improve expression of loading, weight and name from 加载, 重量, 名 to 加载中, 体重, 名称 to stick closer to Chinese expression habits
- fix hard-coded text overall change in widget `MeasurementOverallChangeWidget`, add relevant localization keys in `app_en.arb` and `app_zh.arb`
- fix hard-coded text in widget `Indicator`'s text field, also add relevant localization keys in `app_en.arb` and `app_zh.arb`
- fix hard-coded text in widget `PlanForm`'s `DropDownMenu` by adding a new method named `getI18nLabel` in widget `forms` to get localized label text
- add missing `textPromptTitle` and `textPromptSubheading` keys and their values in `app_zh.arb`

some snapshots after fix:

Link to the issue :

- Link : [Some texts are hard-coded, it is necessary to implement localization #284](#)

Tests

- Since no bugs fixed or feature added, no test cases are added.
- The localization made runs fine on my machine in Chinese environment.
- If there needs to be some kind of tests, please let me know.

Checklist

Please check that the PR fulfills all requirements listed below by checking the relevant checkboxes ([x]). This will ensure a smooth and quick review process.

- ☒ Set a 100 character limit in your editor/IDE to avoid white space diffs in the PR
- ☒ Tests for the changes have been added (for bug fixes / features)
- ☐ Since no bug fixes, no test has been added. The fixed localization works fine on my machine.
- ☒ Added yourself to AUTHORS.md
- ☒ Updated/added relevant documentation (doc comments with `///`).
- ☒ add update logs to relevant descriptions in `app_en.arb` and `app_zh.arb`
- ☒ Added relevant reviewers.

Commit History

- FutureYL3 added 2 commits 1 hour ago
- fix hard-coded text in Chinese version, improve weight expressed in C... f2bc07d
- chore: add Xianglin Zeng to AUTHORS.md and update relevant doc 27e28e7

Workflow Status

- 1 workflow awaiting approval
- This workflow requires approval from a maintainer. [Learn more about approving workflows.](#)
- This branch has no conflicts with the base branch
- Only those with [write access](#) to this repository can merge pull requests.

Add a comment

Write | Preview

Add your comment here...

Markdown is supported | Paste, drop, or click to add files

Close pull request | Comment

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

ProTip! Add `.patch` or `.diff` to the end of URLs for Git's plaintext views.

阐述解决所选 issue 的思路，列出核心代码。

核心思路：使用 flutter_localizations 和 intl 包，配合 arb 文件进行字符串替代以实现语言国际化。

具体思路：

- 修改 app_en.arb 和 app_zh.arb 文件的相关键和值以添加国际化内容
- 在需要国际化的位置使用 AppLocalizations.of(context).key_name 来进行国际化替代
- 在无法直接获得 context 的位置（比如一个只负责与数据交互，不负责与 UI 交互的类中），新建方法传入上下文 context 以调用 AppLocalizations 来进行国际化。

核心代码：

国际化内容

```
"overallChangeWeight": "总体变化",
"@overallChangeWeight": {
  "description": "Overall change in weight"
},
"goalTypeMeals": "从饮食出发",
"goalTypeBasic": "基础",
"goalTypeAdvanced": "进阶",
"chartAllTimeTitle": "{name} 历史记录曲线",
"@chartAllTimeTitle": {
  "description": "All-time chart of 'name' (e.g. 'weight', 'body fat' etc.)",
  "type": "text",
  "placeholders": {
    "name": {}
  }
},
"indicatorRaw": "原始值",
"indicatorAvg": "平均值",
"textPromptTitle": "准备就绪？ ",
"@textPromptTitle": {
  "description": "Title for the text prompt"
},
"textPromptSubheading": "点击右下角按钮开始",
"@textPromptSubheading": {
  "description": "Subheading for the text prompt"
}
```

国际化替代代码：

```
// ignore: prefer_interpolation_to_compose_strings
return Text(AppLocalizations.of(context).overallChangeWeight +
  '$prefix ${delta.abs().toStringAsFixed(1)} $_unit');

Indicator(color: Theme.of(context).colorScheme.primary, text:
AppLocalizations.of(context).indicatorRaw, isSquare: true),
Indicator(color: Theme.of(context).colorScheme.tertiary, text:
```

```
AppLocalizations.of(context).indicatorAvg, isSquare: true),
```

国际化替换方法:

```
String getI18nLabel(BuildContext context) {  
  switch (this) {  
    case GoalType.meals:  
      return AppLocalizations.of(context).goalTypeMeals;  
    case GoalType.basic:  
      return AppLocalizations.of(context).goalTypeBasic;  
    case GoalType.advanced:  
      return AppLocalizations.of(context).goalTypeAdvanced;  
  }  
}
```

4 小结

1. 对本次实验过程和结果的思考

本次实验是为了让我们通过实际操作来了解和参与开源项目的开发流程，包括学习 Markdown 语法、分析开源项目、贡献代码等。以下是对实验过程和结果的一些思考：

1) Markdown 语法学习：Markdown 作为一种轻量级的标记语言，对于文档编写和项目文档的维护非常重要。通过实验内容 1 的学习，我掌握了基本的 Markdown 语法，这对于我未来在技术文档编写和项目文档管理中有大用处。

2) 开源项目参与：实验内容 2 要求我们实际参与开源项目，这不仅让我们了解了开源项目的运作方式，还提升了我们的编程实践能力。通过分析项目结构、参与贡献要求、项目的协议等，我们能够更深入地理解开源项目的运作机制。

3) 团队合作：实验要求不超过 4 人的小组团队完成，这培养了我们的团队合作能力。在实际工作中，团队合作是非常重要的技能，通过这样的实验，我们可以在实践中学习如何与他人协作。

4) 实际贡献：通过选择项目的 issue 或自行提出 issue，并提交 PR，我们体验了从问题识别到问题解决的完整流程，这对于我们理解软件开发生命周期非常有帮助。

2. 对本次实验的收获

1) 技术能力提升：我通过实验学习了 Markdown 语法，提升了文档编写能力；通过参与开源项目，提升了编程和问题解决能力。

2) 团队协作能力：在小组合作中，我们学会了如何与他人沟通和协作，这对于我们未来的学习和工作都非常重要的。

3) 项目管理知识：通过分析开源项目，我们了解了项目管理的基本知识，包括项目结构、协议、活跃度等。