

云存储

如果您的应用程序托管在App Engine上，则可以使用云存储服务，用户可以保存，加载，共享和发布其程序。

设置App Engine

第一个目标是在App Engine上运行您自己的Blockly副本。

1. 下载并安装Python SDK。
2. 登录Google App Engine并创建一个应用程序。
3. 编辑appengine/app.yaml应用程序ID并将其更改为blockly-demo您在上一部中创建的应用程序名称。
4. 将以下文件和目录复制（或软链接）到appengine/static/：
 - demos/
 - msg/
 - media/
 - *_compressed.js
5. 可选：如果您想blockly_uncompressed.js在服务器上使用，还要将该文件appengine/static/复制到appengine/static/，复制closure-library/到父目录中，然后复制到父目录中appengine/。
6. 可选：如果您想运行Blockly游乐场，你必须复制blocks，generators和tests目录，以及文件在第5步。
7. 从GUI运行Google App Engine Launcher，将您的appengine目录添加为现有应用程序，然后按“部署”按钮。如果您更喜欢使用命令行，请运行：appcfg.py --oauth2 update appengine/。

上传Blockly后，您可以将浏览器指向您在步骤2中创建的URL。您应该会看到演示列表，包括云存储演示。

云端交互

在demos / storage / index.html中检查存储演示的来源，并注意以下功能。首先，有一个脚本包含加载云存储API：

```
<script src="/storage.js"></script>
```

请注意，此脚本假定页面上只有一个Blockly工作区。还有这些消息定义，您应该根据需要进行修改：

```
BlocklyStorage.HTTPREQUEST_ERROR = 'There was a problem with the request.\n';  
BlocklyStorage.LINK_ALERT = 'Share your blocks with this link:\n\n%1';  
BlocklyStorage.HASH_ERROR = 'Sorry, "%1" doesn\'t correspond with any saved  
Blockly file.';  
BlocklyStorage.XML_ERROR = 'Could not load your saved file.\n' +  
    'Perhaps it was created with a different version of Blockly?';
```

可以在json目录中的Blockly Games中找到这些消息到其他语言的翻译。

保存当前块只需一次调用BlocklyStorage.link()：

```
<button onclick="BlocklyStorage.link()">Save Blocks</button>
```

要在页面加载时恢复已保存的块，只需在注入Blockly后使用URL的哈希调用BlocklyStorage.retrieveXml:

```
if ('BlocklyStorage' in window && window.location.hash.length > 1) {  
  BlocklyStorage.retrieveXml(window.location.hash.substring(1));  
}
```

本地存储

该storage.js API还提供了在浏览器的本地存储保存一组块的能力。这可以在本地实现而不是在云存储中，或者另外与云存储一起实现（尽管在后一种情况下，必须小心尝试一次恢复的两种类型的存储）。

要从本地存储中恢复块，请在注入Blockly后立即调用BlocklyStorage.restoreBlocks。

```
window.setTimeout(BlocklyStorage.restoreBlocks, 0);
```

要在用户离开页面时自动将块备份到本地存储中，请调用BlocklyStorage.backupOnUnload，它将在页面的unload事件上注册事件侦听器。

```
BlocklyStorage.backupOnUnload();
```