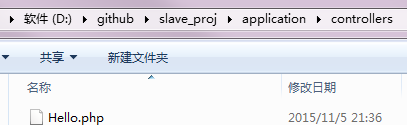
1. Hello world

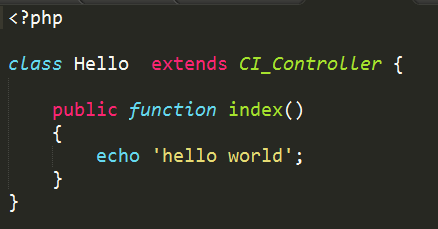
git clone <https://github.com/speedscorpion/slave_proj>



在application/controllers目录下新建Hello.php

注意，Hello首字母必须大写。这个给差评，我郁闷了半天

编辑，粘入以下代码



git add --all

git commit -m hello

git push

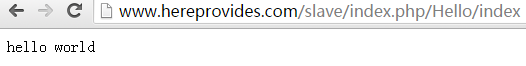
登上服务器120.24.69.137 root 54Xiaosage

cd /var/www/html/slave

git pull

至此，编写部署完成

访问 <http://www.hereprovides.com/slave/index.php/Hello/index>



大家辣么聪明，肯定不用我再多说url为什么是这个了吧

这里的Hello大小写倒是无所谓，再次给差评。url本不应该大小写无所谓

文档在这里<http://codeigniter.org.cn/user_guide/>

好了，怎么开发就到这里，剩下的大家都不成问题了。

数据库的配置，在/application/config/database.php已经配好了

要加数据表就自己加，用客户端连数据库也可以，已授权

直接去服务器上写代码也可以，记得push上github就好了

1. 请求

我们有哪些请求呢

用户进入游戏（凌光）

/Player/enter

没有cookie，写cookie，有cookie则读

cookie中包含的信息

uuid作为用户id

用户昵称，用来展示

是的，我们不需要密码

抓取奴隶（何明）

/Slave/capture

抓取成功，对方成为奴隶，可选择解散或接收对方的奴隶

抓取失败，成为对方奴隶，自己奴隶解散

注意状态的同步

起义（何明）

/Slave/raise

增加到一定人数则起义成功，回到自己的城堡

而其他人下一次登陆时或发起请求时，才会得到解放的信息。web的结构注定了是这样

拉取已起义的人的列表

怀疑（何超）

/Owner/asset

拉取奴隶列表（这个接口其实在作为奴隶主登陆时就可能调用）

/Owner/suspect

确定怀疑对象，成功则将奴隶吊打，同时奴隶那边，下一次登陆出现吊打信息。失败则返回冷冻信息

全局按钮来两个（何超）

/User/show

用来获取图片与链接，用来发圈儿，因为没有开放平台，不能接上微信，出此下策

对应的链接为/share/state

另一个点击之后可以刷新状态

/User/update

其中有些参数是需要从配置文件读取的。这个配置的读写由凌光同学负责

<root>

<strategy select=”1”/>

<config index=”1”>

<item1 value=””/>

…

</config>

<config index=”2”/>

<item1 value=””/>

…

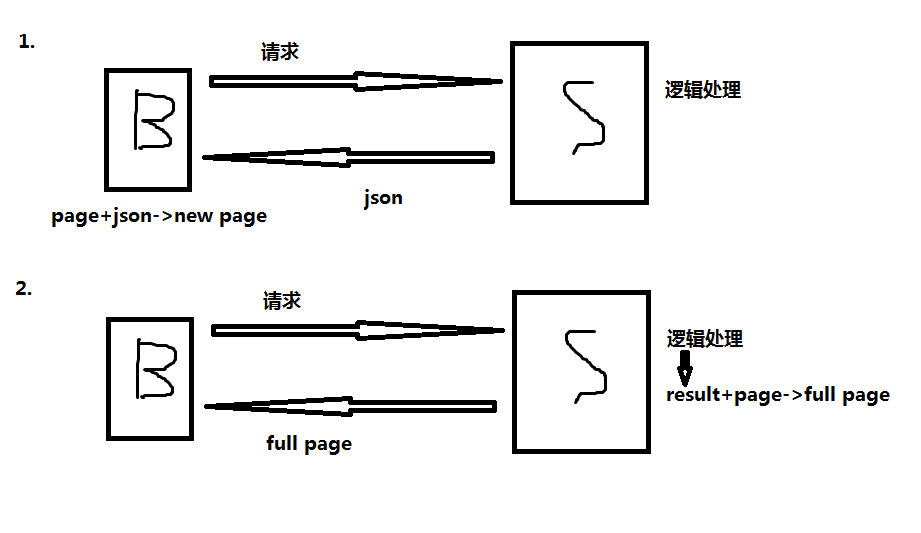
</config>

…

</root>

大概就是这样，要保证改变一项，就能换一套配置。

1. 请求过程



有两种，区别在于，第一种，逻辑处理得到的数据直接返回，完整的页面在浏览器上由广森合成。第二种，逻辑处理得到的数据在服务器合成，返回完整地网页。

我写的示例中的就属于第一种情况。在实际的开发中，我们统一采用json格式传输。

第二种请求中的page仍然有广森写。后台同学只需要把逻辑结果填进去，具体怎么做，示例可以在官方文档中找到。

辣么，怎么选择用哪一种呢？很简单，当整个页面都需要更新的时候，即场景要跳转，用第二种，否则第一种即可完成。这样做的目的是减少请求次数，减少请求的数据量。

1. 场景

初始场景:每个人呆在自己的城里。一个城堡的画面，就这样。

被奴役:被其它人奴役。一个监狱铁栏杆，人流鼻涕扒着的画面。

发起起义:按钮，用来起义。列表，一起起义的人名列表。监狱里人群涌动的画面。

镇压:按钮，怀疑有人造反。列表，选择你怀疑的人。怀疑对了可以吊起来打。怀疑错了冷冻一段时间不得再怀疑。

起义失败:被吊起来的画面。一段时间内，不得造反。

起义成功:回到自己城堡的画面。点击城堡，可以看到之前的奴隶主，对，她/他现在是你的奴隶。

全局按钮:炫耀自己的奴隶，哭诉自己的遭遇。所以我们还需要一个页面用来向其它人展示炫耀或惨剧的效果页。这个页面不需要是玩家也可以访问，并给出加入游戏的链接。刷新页面按钮

另

前端由广森完成。

潘博渊同学负责写文档跟ppt，有问题与我沟通。

请后台开发同学注意，辛苦一下，周六之前，完成这个框架的体验。具体做法，从数据库读出数据，然后合成页面返回。然后给出工作量评估，预期什么时候完成自己的接口。微信上发给我。

梁昊同学下周三之前给出测试用例。