

# 1 系统界面

如图 1.1-1.4 所示为系统界面，包括 Home 页面，Features 页面（Health Check 和 Performance Compare）和 Contact 页面。

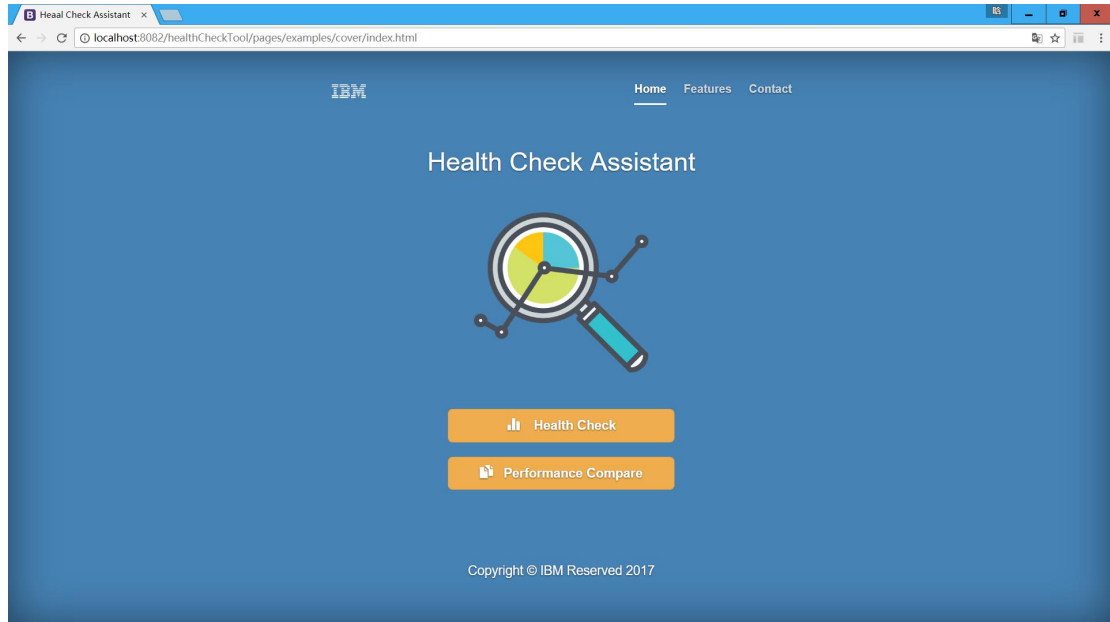


图 1.1 首页

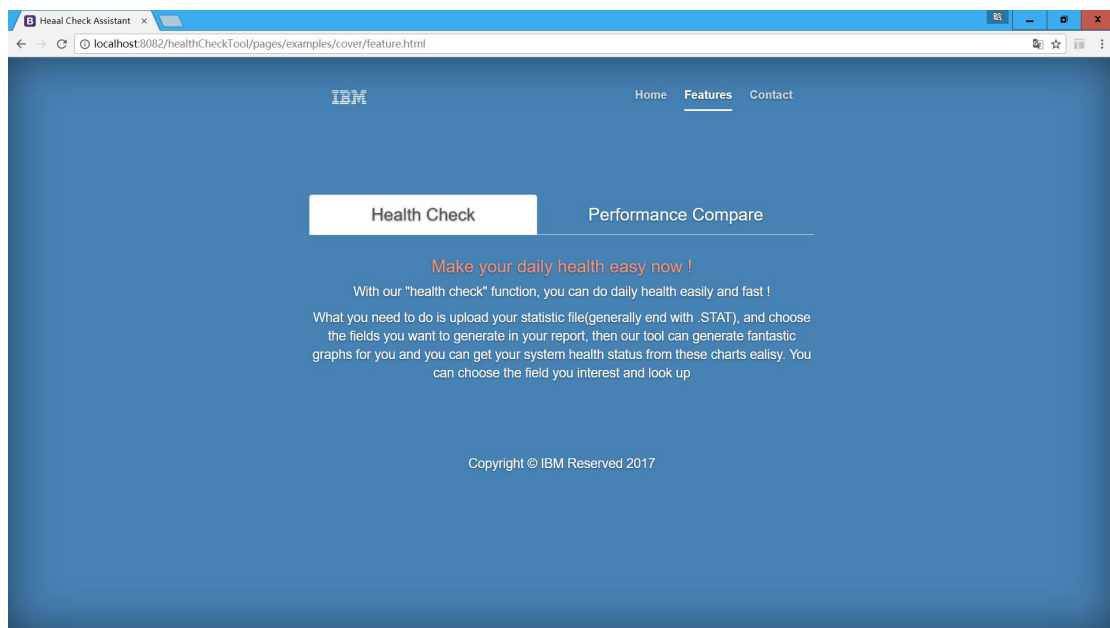


图 1.2 HealthCheck Features

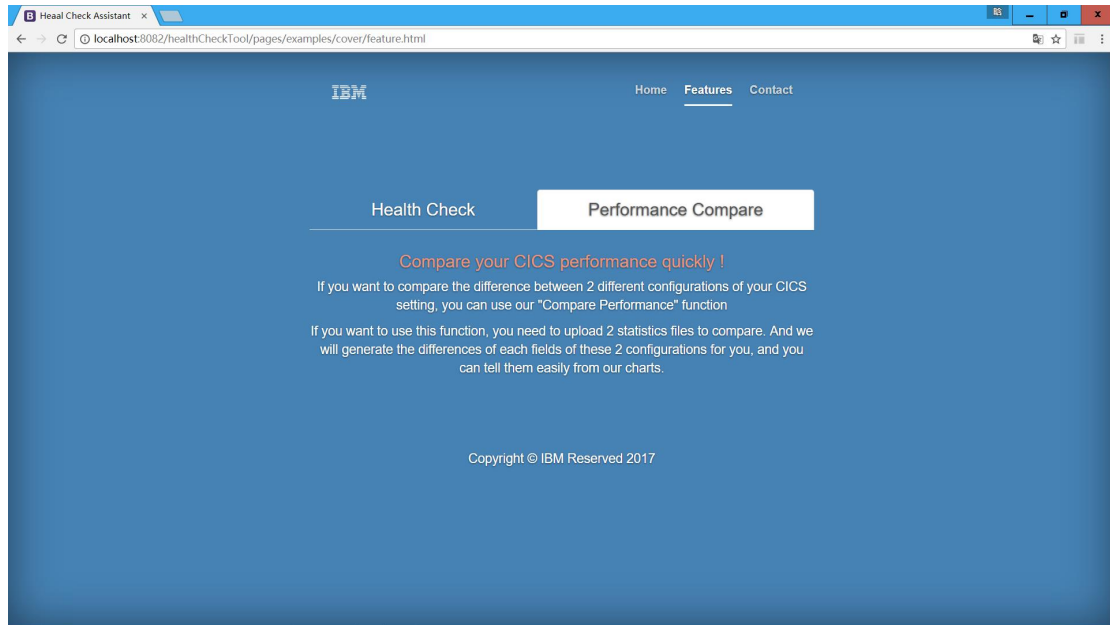


图 1.3 Performance Compare Features

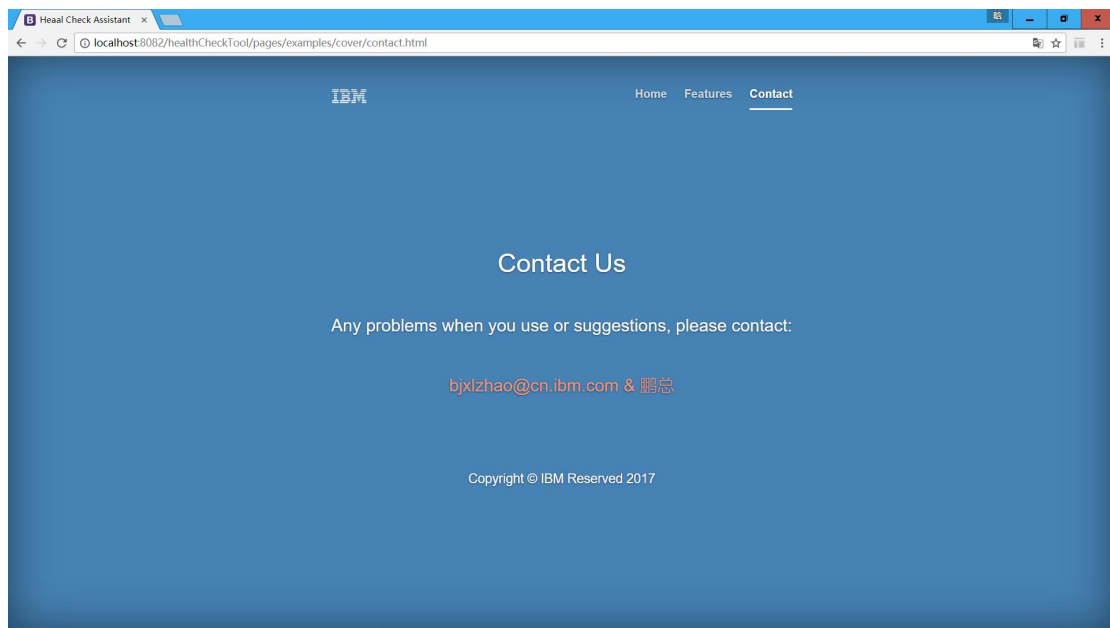
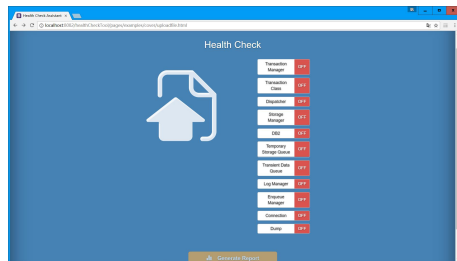
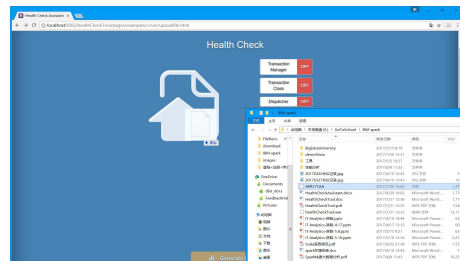


图 1.4 Contact

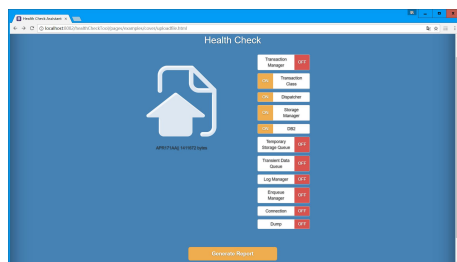
## 2 健康检查



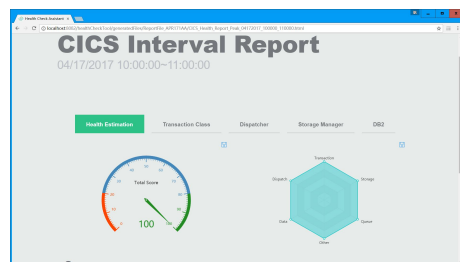
a 初始状态



b 拖拽上传



c 生成报告



d 显示结果

图 2.1 健康检查

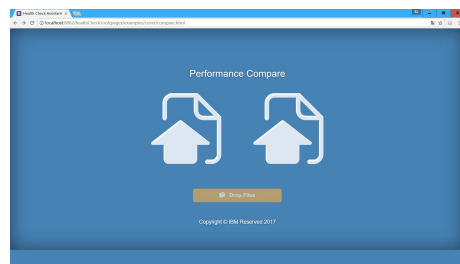
如图 2.1 所示，采用拖拽上传方式用 `ajax+formData` 上传日志文件，后台对上传的日志文件用 `python` 脚本处理，生成健康检查报告。

- a 为初始状态，`button` 为不可点击状态。
- b 为拖拽上传，拖拽之后直接上传文件。
- c 为用户选择要显示的模块，点击 `button` 生成报告。
- d 为调用 `python` 脚本进行健康检查，生成的结果。

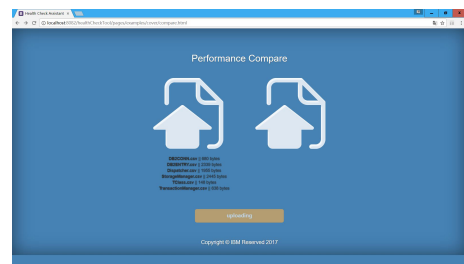
上次所说的 `Exited with error code 1` 错误。现在表现为上传 1379KB 的 `APR171AA` 文件可以上传成功。上传 582MB 的 `Statistics_sample` 文件没有上传成功。

猜测可能原因为浏览器上传限制或服务器限制或机器内存限制。

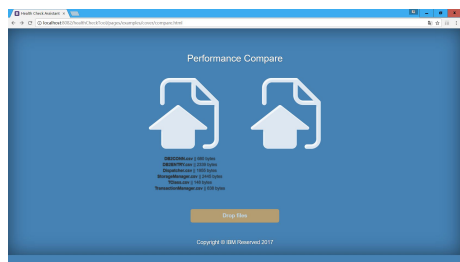
### 3 配置对比



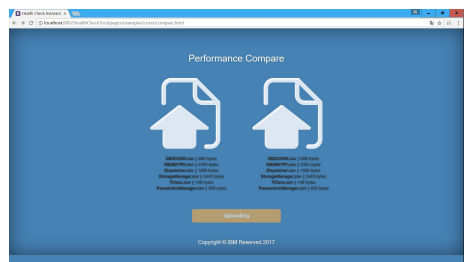
a 初始状态



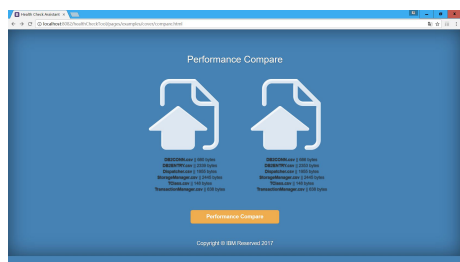
b 上传第一个文件夹



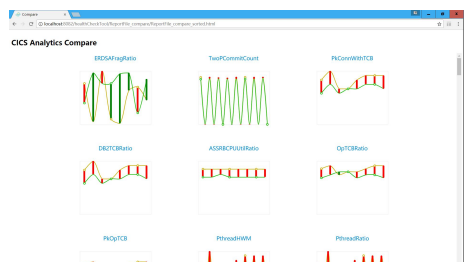
c 上传完成



d 上传第二个文件夹



e 上传完成



f 显示结果

图 3.1 配置对比

如图 3.1 所示，为配置对比页面。通过拖拽的方式将配置前和配置后的文件分别拖拽进行上传。后台调用 `python` 脚本进行对比操作，生成比较结果。拖拽文件夹会采用递归的方式检索文件夹内的 `rawData` 文件夹。拖拽完成后自动上传。上传位置为 `WebContent` 下的 `DATA` 文件夹。以“`user+进入页面的时间戳`”命名。内部用 `scene1` 和 `scene2` 分别存储两个文件夹的内容。

a 为初始状态，button 内容为“drop files”，且不可点击。

b 为拖拽第一个文件夹后的状态，显示文件夹内容，button 内容为“uploading”。且不可点击。

c 为第一个文件夹上传完成的状态，button 显示“drop files”且不可点击。

d 为拖拽第二个文件夹后的状态，显示文件夹内容。Button 内容为“uploading”。且不可点击。

e 为第二个文件夹上传完成的状态，button 显示“Performance Compare”。可以点击。

f 为点击 button 后显示的对比结果。

## 4 下一步工作

1. 大文件上传问题
2. 是否需要点击弹出文件选择框上传
3. 目前每次都需要上传运行， 采取 保留历史记录 or 下载健康报告 哪种方式
4. 服务器需不需要存储历史数据