

1. Система ранжирования

2. Зависимости

- C++14
- UjsonCpp - библиотека для работы протокола (repo: <https://github.com/NightCodersss/>)
- cppunit - библиотека для unit-тестирования (используется в Ujson)
- json - библиотека для чтения конфига (repo: <https://github.com/nlohmann/json>)

3. Разработчику

Помимо информации, изложенной здесь, следует изучить документацию, сгенерированную doxygen'ом (директории latex и html в корне проекта).

3.1. Структура

Структура выглядит так: (Фронт-энд или другой сервис) $\xrightarrow{\text{Southernprotocol}}$ Ranking System $\xrightarrow{\text{Northernprotocol}}$ IndexServer Где ' \rightarrow ' обозначает подключение от клиента к серверу.

3.2. Протокол

В связи с тем, что Ujson парсится на лету, не все поля обязательны. В силу гибкости протокола незначительное изменение или значительное добавление не должно требовать больших трудов. Что такое южный и северный легко понять отсюда ??.

3.2.1. Южный

Необходимо передавать строчку query, которая хранит в себе запрос.

```
{  
  "query": "what is ...?"  
}
```

Планируется добавить параметры, каким образом представлять информацию о документе обратно.

Заглушка работает так: слушает некоторый порт, по приходу данных, до переноса строки ($\backslash n$), приходящие данные пакует в ubjson и отправляет системе ранжирования. По приходу данных от системы ранжирования (то есть системы поиска)(в ubjson) заглушка "их отвечает" в json (обычном).

3.2.2. Северный

Такой же, как "южный" но с параметром - какой индекс использовать (по какому тексту), то есть:

```
{
  "query": "what is ...?",
  "index_id": "aa1234df",
  "fields": ["docname", "author", "smallpartoftext"],
}
```

3.2.3. Реализация

Ясно, что нехорошо ждать каждого ответа. Глобально, есть два пути решения этой проблемы: использование асинхронных операций ввода-вывода (с callback'ами) и использование большого количества потоков. Существенных различий нет (нам известных), за исключением того, что код с большим количеством callback'ов плохо читаем и некрасив.

Итак, реализация такова, что на каждое соединение выделяется отдельный поток, в котором соединение работает синхронно.

4. Программистам-пользователям

См. структуру ??

5. Человеку, который запускает

См. структуру ??

5.1. Конфигурация

Пример конфига RankingServer'a:

```
{
  "texts": [
```

```
{
  "servers": [{"host": "localhost", "port": "14000"}],
  "factor": 1.8,
  "name": "The first text field",
  "index_id": "1",
},
{
  "servers": [{"host": "", "port": "14000"}],
  "factor": 0.9,
  "name": "Useless information",
  "index_id": "100400ab",
},
]
}
```