# Задание

Необходимо написать информационный сервер крупной научной конференции, проходящей одновременно в нескольких городах.

Используемые технологии: ASP.NET (можно ASP.NET Core), все остальное по собственному выбору.

Общая схема работы: API предоставляет возможность регистрировать, хранить и предоставлять доступ по запросу различные секции конференций. Web-интерфейс позволяет просматривать список всех зарегистрированных секций и регистрировать новые секции.

#### API

API центрального сервера состоит из следующих методов:

/conference/<section>/info PUT, GET

/conference/info GET

Все данные передаются в формате JSON. Каждый метод проиллюстрирован примером корректного запроса и ответа.

Структура JSON-а будет ясна из примеров: можно считать, что все указанные в примере поля должны содержаться в корректном запросе и других полей там не будет. Типы полей также однозначно определяются по примерам. Все строковые значения могут состоять из произвольных unicode-символов.

Если в описании метода API не указано тело запроса, значит оно должно быть пустым. Если не указано тело ответа, предполагается пустой ответ с кодом 200 OK.

#### Приём данных от серверов секций конференции

PUT /conference/<section>/info (advertise-запрос)

Запрос:

{

"name": "Geoinformation Systems",

"city": "Tomsk",

"location": "Lenina 2, 404",

}

Здесь **section** является уникальным идентификатором сервера: при получении нового advertise-запроса с тем же endpoint-ом информация перезаписывается. **section** имеет вид аббревиатуры названия секции **<GIS>**.

Результаты от серверов, не приславших advertise-запрос, не должны сохраняться. Таким серверам нужно отвечать пустым ответом с кодом 400 Bad Request.

#### Получение текущей информации о серверах конференции

GET /conference/<section>/info

Ответ:

Этот метод должен вернуть последнюю версию информации, полученную PUT-запросом по этому адресу в том же формате. Если сервер с таким **section** никогда не присылал advertise-запрос, нужно вернуть пустой ответ с кодом 404 Not Found.

GET /conference/info

Ответ:

[

{

"section": "GIS",

"info": {

"name": "Geoinformation Systems",

"city": "Tomsk",

"location": "Lenina 2, 404",

}

},

{

"section": "CS",

"info": {

"name": "Computer Science",

"city": "Tomsk",

"location": "Lenina 30, 206",

}

}

}

]

Ответ должен содержать последнюю версию информации о всех серверах, когда-либо присылавших advertise-запрос.

#### Web-интерфейс

Назовём это творческим заданием, особых требований выдвигать не будем, ограничений тоже.

По сути требование одно: Web-интерфейс должен позволять зарегистрироваться на конференции и получить информацию серверах конференции.