Разработать сервис для извлечения данных о рынках баскетбола на сайте bet365.com анализом трафика с помощью промежуточного прокси. Решение должно быть написано на TypeScript и запускаться на node.js. Нужно разобрать формат передачи данных, который использует bet365. И отдавать данные в требуемом виде.

Environment Variables:

* MAIN\_PROXY=1.2.3.4:1080 – основной socks5 прокси который нужно использовать для доступа в интернет

Компоненты решения

* local-proxy – socks 5 прокси, нужно разработать
* main-proxy – существующий socks 5 прокси, через который в итоге должен идти весь трафик
* chrome – будет настроен на работу c local-proxy и для работы с самоподписанным сертификатом (chrome НЕ должен запускаться программно, эта задача находится за пределами данного сервиса, для тестирования можете использовать скрипт или расширение для работы с прокси). При запуске он будет открывать <https://www.bet365.com/#/IP/B18> В chrome будет открыта только одна вкладка
* premws-local – http сервер который будет принимать websocket по адресу /zap/?uid=<uid> Затем открывать WebSocket к реальному wss://premws-pt\*.365lpodds.com/zap/?uid=<uid> используя при этом main-proxy. Обрабатывать сообщения об обновлениях рынка и отправлять обновленные событитя клиентам bet365-updates.
* bet365-updates – WebSocket сервер написанный с использованием <https://www.npmjs.com/package/ws> он должен посылать WebSocket сообщение, а конкретно JSON с состоянием события вместе со всеми рынками

Задача сводится к разработке local-proxy, premws-local, bet365-uptades в виде единого сервиса

План работы

* Создать local-proxy который будет слушать 127.0.0.1:1080
* По умолчанию перенаправлять трафик на process.env["MAIN\_PROXY"]
* Дальше будет запущен chrome с флагом –proxy-server=127.0.0.1:1080 –lang=en-GB
* Сгенерировать самоподписанный сертификат для \*.365lpodds.com
* Добавить сертификат в список доверенных, чтобы chrome не ругался на сертификат
* Создать https сервер, например на порту 8090, использовать сгенерированный сертификат
* Перенаправлять прокси трафик, идущий на premws-pt1.365lpodds.com на порт 8090
* Принимать websocket по адресу /zap/?uid=<uid>
* Открывать WebSocket к реальному wss://\*.365lpodds.com/zap/?uid=<uid>
* Перенаправлять все сообщения прозрачно в обе стороны

План не является полным и предоставлен чтобы лучше понять требования. Возможно, что сообщений из wss://premws-pt\*.365lpodds.com/zap/?uid=<uid> будет недостаточно для инициализации состояния сервиса и может быть нужно будет перехватывать запросы на bet365.com, для чего нужно будет добавить соответствующий сервер (+сертификат) и правило для перенаправления прокси на него. Также возможны некоторые вариации, к примеру если есть возможность перехватывать данные без создания лишних промежуточных серверов, сразу средствами прокси, то лучше так и сделать, однако это может потребовать слишком большого количества работы и вариант с промежуточными http серверами является приемлемым

WebSocket сервер с обновлениями (bet365-updates)

При каждом подключении клиента к WebSocket сервис должен отправить N сообщений, каждое сообщения представляет из JSON объекта имеющий тип EventState (смотреть далее). Где N – это количество активных событий в баскетболе

После изменения любого параметра события (включая рынки) сервис должен посылать объект события целиком всем клиентам, таким образом сервис посылает только объекты событий сериализированных в JSON.

На данный момент интересуют рынки следующих типов:

* Money Line
* Double Chance
* Spread
* Totals
* Team totals

Тип объекта EventState:

export type EventState = {

    id: string,

 sport: "basketball",

    league: string,

    team1: string,

    team2: string,

    clock: string, // "01:23"

    clockState: "paused" | "running",

    period: "Q1" | "Q2" | "Q3" | "Q4" | "1st" | "2nd" | "OT"

    markets: Array<Market>

}

export type Market = {

    type: 'win 3-way',

    context: MarketContext,

    side: 'team1' | 'team2' | 'draw',

    odds: number, // OD

    suspended: boolean, // SU

    extra: Bet365Extra,

} | {

    type: 'win 2-way',

    context: MarketContext,

    side: 'team1' | 'team2',

    odds: number,

    suspended: boolean,

    extra: Bet365Extra,

} | {

    type: 'double chance',

    context: MarketContext,

    side: 'team1 or team2' | 'team1 or draw' | 'draw or team2',

    odds: number,

    suspended: boolean,

    extra: Bet365Extra,

} | {

    type: 'totals over/under',

    context: MarketContext,

    designation: 'over' | 'under',

    points: number,

    odds: number,

    suspended: boolean,

    extra: Bet365Extra,

} | {

    type: 'totals 3-way over/under/exact',

    context: MarketContext,

    designation: 'over' | 'under' | 'exact'

    points: number,

    odds: number,

    suspended: boolean,

    extra: Bet365Extra,

} | {

    type: 'totals for team over/under',

    context: MarketContext,

    designation: 'over' | 'under'

    side: 'team1' | 'team2',

    points: number,

    odds: number,

    suspended: boolean,

    extra: Bet365Extra,

} | {

    type: 'totals for team 3-way over/under/exact',

    context: MarketContext,

    designation: 'over' | 'under' | 'exact'

    side: 'team1' | 'team2',

    points: number,

    odds: number,

    suspended: boolean,

    extra: Bet365Extra,

} | {

    type: 'spread',

    context: MarketContext,

    side: 'team1' | 'team2',

    points: number,

    odds: number,

    suspended: boolean,

    extra: Bet365Extra,

} | {

    type: 'spread 3-way',

    context: MarketContext,

    side: 'team1' | 'team2' | 'draw',

    points: number,

    odds: number,

    suspended: boolean,

    extra: Bet365Extra,

}

export type Bet365Extra = {

    ID: string,

    FI: string,

    OD: string

}

export type MarketContext = {

    type: 'match',

    time: 'none' | 'regular' | 'with overtime'

} | {

    type: 'half',

    half: number,

    time: 'none' | 'regular' | 'with overtime'

} | {

    type: 'quarter',

    quarter: number,

    time: 'none' | 'regular' | 'with overtime'

}

Дополнительные требования

* Перезагрузка страницы не должна приводить к рассинхронизации
* Должно корректно обрабатываться появление нового спортивного события
* Должно корректно обрабатываться завершение спортивного события (должно удаляться из списка активных событий)
* bet365.com будет открываться на английском языке, соответственно сервис должен ожидать что сообщения premws-pt\*.365lpodds.com также будут на английском

Расшифровка некоторых сокращений wss://premws-pt\*.365lpodds.com/zap/?uid=<uid>

|  |  |
| --- | --- |
| U | Update |
| D | Delete |
| F | Snapshot |

<http://help.bet365data.com/bet365/FieldsExplanation.html>