Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут» імені Ігоря Сікорського
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Лабораторна робота № 6

З дисципліни: «Компоненти програмної інженерії – 2.
Моделювання та аналіз програмного забезпечення»

Виконав:
студент групи ІТ-73
Похиленко Олександр Андрійович
Перевірив:
Галушко Д. О.

Київ 2020

Тема: Розробка конструктора Telegram-ботів для замовлень.

Мета: Обрати спосіб передачі даних про розмітку бота, а також завантаження та обробки цих даних.

В конструкторі ботів дерево розмітки грає дуже велику роль. У минулих лабораторних вже були обрані основні інтерфейси та класи вузлів, що мають пряме відношення до цього дерева. Тепер необхідно обрати спосіб передачі цього дерева. При цьому необхідно розуміти, що це дерево буде редагуватися користувачами через веб-сайт, зберігатися в базу даних та оброблюватися на стороні сервера ботів під час запуску.

Серед різних форматів обміну даними був обраний JSON, оскільки він дуже просто оброблюється за допомогою JavaScript на стороні користувача у браузері, не потребуючи додаткових бібліотек. Крім цього, у нашому випадку використовується PostgreSQL, де присутня можливість працювати з JSON, хоча й вона поки не використовується (але може бути використана у майбутньому). На стороні сервера ботів використовується Newtonsoft Json.NET, що дозволяє зручно працювати з отриманими даними за допомогою LINQ to JSON.

Коли користувач відкриває редактор бота, то він отримує дерево розмітки у форматі JSON, яке відновлюється за допомогою написаної функції loadFromJSON. Більш детально про редактор бота на стороні клієнта буде написано в лабораторній роботі №7. Коли ж користувач хоче зберегти відредаговане дерево, то за допомогою функції sendToServer дані відправляються серверу та зберігаються в базі даних у «чернетку». Якщо ж у цей момент бот працює, то він не зупиниться, а продовжить працювати на старій версії розмітки, поки його не зупинять. Тільки коли користувач вирішить перезапустити бота, сервер ботів дістане цю розмітку з бази даних і перевірить її на коректність перед запуском.

Серіалізована розмітка дерева, насправді, є не деревом, а масивом вузлів, які були отримані обходом дерева у глибину. Це необхідно для зручності, оскільки саме у цьому випадку можна застосувати LINQ to JSON на стороні сервера ботів. Крім цього, можливо, що у майбутньому буде відправлятися лише масив змінених вузлів, що дозволить зменшити навантаження на мережу.

Кожен вузол має у собі **ідентифікатор** та об'єкт, що описує його **параметри**. Усі вузли, крім кореневого, ще й мають **ідентифікатор батьківського вузла**, щоб на стороні сервера була можливість знову зібрати масив у дерево. Ідентифікатори вузлів генеруються клієнтом інкрементно під час обходу дерева перед відправкою, тож кореневий вузол буде завжди мати id = 0, а його перший дитячий вузол id = 1.

Кожен об'єкт параметрів вузла має у собі **ідентифікатор типу**, що вказує на тип вузла, які були перераховані в лабораторній роботі №5. Тип вузла дозволяє на стороні сервера зрозуміти, у який саме клас необхідно десеріалізувати отриманий вузол. Крім цього, будь-який об'єкт параметрів має у собі поля для **назви** вузла, його текстового **повідомлення**, а також для **ідентифікатора файлу** Telegram і його **прев'ю**. Інформаційний вузол не має ще якихось додаткових параметрів.

Вузол розділу має у собі ще й параметр, що вказує на **тип колекції** (блочний вузол чи сторінковий). Цей тип вказується просто числом, яке зіставляється з необхідним типом класу на стороні сервера. Так, наприклад, блочний вузол розділу перетвориться на BlockNode, а сторінковий – на FlipperNode.

Товарний вузол має так само параметр, що вказує його **вигляд** (дерево зі звичайних вузлів чи один мультивузол). Для підтримки підтипів товарів й можливості встановлювати кожному підтипу свою ціну також додані й інші параметри. Параметр **характеристик** містить у собі масив об'єктів, що описують кожен підтип (його ***назву***, ***список підвидів***, а також може містити

додаткове текстове ***повідомлення*** і ***файл***). Далі йде параметр **значень цін**, що представлені у вигляді масиву. Кількість елементів масиву повинна дорівнювати добутку усіх підвидів кожного підтипу. Якщо ж елементів менше чи більше, то під час десеріалізації користувач буде повідомлений про пошкоджені дані.

Вузол вводу даних має тільки одну відмінність від стандартних параметрів – він має у собі параметр, що вказує на **тип введених даних**, які будуть оброблюватися цим вузлом. Цей тип вказує, на який з шести класів (TextInputNode, TimeInputNode, ImageInputNode, AudioInputNode, VideoInputNode чи DocumentInputNode) перетвориться вузол під час десеріалізації.

Вузол відправки замовлення також відрізняється від стандартного тільки одним параметром. Цей параметр є ідентифікатором **групи статусів**, що буде застосовуватися для швидких відповідей на замовлення, зроблених через цей вузол. Про групи статусів та обробку замовлень буде більш детально розписано у лабораторній №8. На наступній сторінці можна побачити приклад розмітки у форматі JSON, де використовуються усі вузли.

```
[
  {
    "id": 0,
    "parameters": {
      "type": 1,
      "name": "Корінь",
      "message": "Вітаємо!",
      "fileId": null,
      "previewId": null
    }
  },
  {
    "id": 1,
    "parentId": 0,
    "parameters": {
      "type": 2,
      "name": "Інформаційний вузол",
      "message": "(Якась інформація.)",
      "fileId": null,
      "previewId": null
    }
  },
  {
    "id": 2,
    "parentId": 0,
    "parameters": {
      "type": 3,
      "name": "Розділ",
      "message": "(Повідомлення сторінкового розділу.)",
      "fileId": null,
      "collType": "2"
    }
  },
  {
    "id": 3,
    "parentId": 2,
    "parameters": {
      "type": 4,
      "name": "Товар",
      "message": "Вкажіть параметри товару:",
      "fileId": null,
      "displayType": "2",
      "properties": [
        {
          "name": "Підтип 1",
          "types": [
            "Підвид 11",
            "Підвид 12",
            "Підвид 13"
          ],
          "message": "Оберіть підтип 1:",
```

```
          "fileId": null
        },
        {
          "name": "Підтип 2",
          "types": [
            "Підвид 21",
            "Підвид 22",
            "Підвид 23"
          ],
          "message": null,
          "fileId": null
        }
      ],
      "values": [
        50,
        60,
        65,
        70,
        80,
        85,
        80,
        90,
        95
      ],
      "previewId": null
    }
  },
  {
    "id": 4,
    "parentId": 0,
    "parameters": {
      "type": 5,
      "name": "Вузол вводу даних",
      "message": "Введіть час (hh:mm).",
      "fileId": null,
      "inputType": "2"
    }
  },
  {
    "id": 5,
    "parentId": 4,
    "parameters": {
      "type": 6,
      "name": "Вузол відправки даних",
      "message": "Ваше замовлення було відправлено.",
      "fileId": null,
      "statusGroupId": "1"
    }
  }
]
```