

ПОРІВНЯННЯ REST І SOAP ТЕХНОЛОГІЙ

В сучасному світі Інтернет є глобальною мережею, що об'єднує безліч комп'ютерів, з'єднаних для спільного використання ресурсів та обміну інформацією. Він сполучає безліч типів мереж, серед яких можуть бути урядові, наукові, мережі малих фірм і великих корпорацій.

Основою такого з'єднання є схема «клієнт – сервер». Надання послуг здійснюється спільною роботою двох процесів: на комп'ютері користувача і на комп'ютері-сервері.

Нашою метою є порівняти та розглянути дві технології REST і SOAP, що базуються на схемі клієнт-сервер, для побудови додатків на основі протоколу HTTP.

Більшість інформації через Internet передається за допомогою HTTP. Раніше цей протокол використовувався лише для пересилання інформації невеликих розмірів. За передачу більших файлів відповідав протокол FTP (FileTransferProtocol). Зараз по HTTP передаються файли довільних розмірів і він є основним протоколом, який за це відповідає. На основі HTTP базується набір принципів проектування, відомий як REST і дозволяє охопити всю потужність шляхом побудови інтерфейсів, що може використовуватися практично з будь-якого пристрою чи операційної системи [1].

REST – це стиль архітектури програмного забезпечення для побудови розподілених масштабованих Web, заснований на маніпулюванні ресурсами і специфікації HTTP. REST ілюструє розвиток Web архітектури, характеризуючи і регулюючи макровзаємодію чотирьох Web компонентів, а саме серверів, мережевих шлюзів, проксі і клієнтів, без застосування обмежень до індивідуальних учасників. Таким чином, REST по суті визначає правильну поведінку учасників [2].

SOAP – це простий протокол (для його використання, немає необхідності в написанні нового), заснований на ідеї, що в деякий момент в розподіленій архітектурі виникає необхідність обміну інформацією. Крім того, для систем, в яких існує ймовірність перевантажень та ускладнень при обробці процесів, цей протокол вигідний тим, що вимагає мінімальної кількості ресурсів. Він дозволяє здійснювати всі операції через HTTP. У специфікації SOAP існує три базових компоненти: конверт SOAP (SOAPenvelope), набір правил шифровки і засоби взаємодії між запитом і відповіддю.

Для більшого розуміння переваг та недоліків, переходимо до порівняння підходів SOAP і REST.

SOAP – це ціле сімейство протоколів і стандартів, і це говорить про його масивність і складність з точки зору машинної обробки. Тому REST працює швидше [4].

SOAP використовують HTTP як транспортний протокол, в той час як REST базується на ньому. Це означає, що всі існуючі напрацювання на базі протоколу HTTP, такі як кешування на рівні сервера, масштабування, продовжують так само працювати в REST архітектурі, а для SOAP необхідно шукати інші засоби. Замість цього SOAP сервіси отримують таку властивість, як можливість працювати з будь-яким протоколом транспортного рівня замість HTTP.

SOAP більш застосовний в складній архітектурі, де взаємодія з об'єктами виходить за рамки теорії CRUD, а ось в тих додатках, які не покидають межі цієї теорії, цілком придатним може виявитися саме REST через свою простоту і прозорість. Дійсно, якщо будь-яким об'єктам вашого сервісу не потрібні більш складні речі, крім: «Створити», «Прочитати», «Змінити», «Видалити» (як правило - в 99% випадків цього достатньо), можливо, саме REST стане правильним вибором. Крім того, REST в порівнянні з SOAP, може виявитися і більш продуктивним, так як не вимагає витрат на розбір складних XML команд на сервері (виконуються звичайні HTTP запити - PUT, GET, POST, DELETE). Хоча SOAP, в свою чергу, більш надійний і безпечний.

Зі сказаного вище можна зробити висновок, що SOAP і REST представляють різні вагові категорії і навряд чи знайдеться завдання, для якого буде складно сказати, який підхід раціональніше використовувати SOAP чи REST. В подальшому планується розробка клієнтської частини з використанням REST підход на мові програмування Javascript.

Список використаних джерел :

1. Самсонов В. В., Эрохин А. Л. Методи та засоби Інтернет-технологій; - Х.: СМІТ, 2008, - 264с.
2. М. Тим Джонс. Використання REST-технологій [Електронний ресурс]/. М. Тим Джонс. - Режим доступу: <http://www.ibm.com/developerworks/ru/library/os-understand-rest-ruby/>.
4. М. Стадник. [Електронний ресурс] / Михаил Стадник. –Режим доступу: <http://mikhailstadnik.com/practical-web-services>