

ჩაწერის ოპერაციათა კოორდინაცია სემანტიკური ჰიპერმიმითებლების მეშვეობით

იოსებ ძმანაშვილი, ზურაბ ბოსიკაშვილი
საქართველოს ტექნიკური უნივერსიტეტი
რეზიუმე

სემანტიკური ჰიპერმიმითებლების(ბმულების) როლი უმნიშვნელოვანესია ჰიპერტექსტზე დაფუძნებული მანქანასა და მანქანათაშორის ურთიერთქმედების მართვისთვის განკუთვნილ სერვისებში. სტატია ეხება სემანტიკური ბმული კავშირების გამოყენების, ახალი ბმული კავშირების შექმნის აუცილებლობისა და არსებული მედია ფორმატების განვრცობის საშუალებებს. გამოყენებული მასალა ეფუძნება RFC6861 სტანდარტის სპეციფიკაციის დამუშავების პროცესში ავტორების მიერ შექმნილ გამოცდილებას.

საკვანძო სიტყვები: ჰიპერმედია სერვისი. ჰიპერტექსტი. ჰიპერმიმითებელი. HTTP მეთოდები. მედია ტიპები. კლიენტ-სერვერი.

1. შესავალი

სხვა ტიპის ვებ სერვისებისგან განსხვავებით, ჰიპერმედია სერვისებში, კლიენტსა და სერვისს შორის არსებულ ურთიერთქმედებათა კოორდინაცია ხორციელდება მეტა- და სამართავი ინფორმაციის მოთხოვნა/პასუხის გზავნილებში ჩართვით. სამართავი ინფორმაცია შესაძლებელია წარმოდგენილი იყოს როგორც HTTP გზავნილის თავსართების, ასევე რესურსების რეპრეზენტაციებში ჩართული ბმულების, ბმული კავშირებისა და სხვა, ისეთი სამართავი ელემენტების მეშვეობით როგორცაა: დასამუშავებელ მონაცემთა გადასაცემი ვებ მისამართი, მონაცემების კოდირების ფორმატი და მოთხოვნის გასაგზავნად აუცილებელი HTTP მეთოდი.

მოცემული მომენტისთვის, ჰიპერმედია სერვისებში მონაცემთა ჩაწერის ოპერაციების კოორდინაცია განეკუთვნება ერთერთ ბუნდოვან დეტალს. ხშირ შემთხვევაში არსებული HTTP მეთოდები გამოიყენება არამიზნობრივად, მათი ძირითადი დანიშნულების გვერდის ავლითა და/ან მათი გამოყენების სხვა ასპექტების იგნორირებით. აღნიშნული პრობლემა გამოწვეულია ჩაწერის ოპერაციების(შექმნა, ჩანაცვლება, მოდიფიცირება) აღწერასა და მათ განსახორციელებლად აუცილებელ HTTP მეთოდებს(POST, PUT, PATCH) შორის კავშირებისა და მათი სემანტიკის განსაზღვრის შეთანხმებული შესაძლებლობის არარსებობით. მოცემული კვლევის ფარგლებში განვიხილავთ რესურსის შექმნის, ჩანაცვლების და ნაწილობრივ მოდიფიცირების ოპერაციებსა და მათ მართვასთან დაკავშირებულ პრობლემატიკას.

მომდევნო ქვეთავებში განვიხილავთ აღნიშნული ოპერაციების აღწერასა და მართვასთან დაკავშირებული საკითხების გადაჭრის გზებს:

- მედია ტიპებისგან დამოუკიდებელი ზოგადი დანიშნულების მკაფიო სემანტიკის მქონე ბმული კავშირების განსაზღვრა ჩაწერის ოპერაციების აღსაწერად;
- არსებული მედია ტიპების განვრცობის საშუალებები, ჩაწერის ოპერაციების მართვის სემანტიკის მქონე ბმული კავშირებისა და სხვა დამხმარე მედია ტიპების მეშვეობით;
- არსებული HTTP მეთოდების ასპექტების სრულყოფილად გამოყენების გზები.

2. არსებული მიდგომები

გავრცელებული პრაქტიკის მიხედვით ჩაწერის ოპერაციების შესასრულებლად გამოიყენება HTTP პროტოკოლის მეთოდები რომელთა გამოყენება ხშირ შემთხვევაში ეფუძნება: ა) ზეპირ შეთანხმებებს; ბ) გარე, ადამიანისთვის გასაგებ ფორმატში შექმნილ დოკუმენტაციას(ე.წ. სერვისის

დოკუმენტაცია); ან გ) მედია ტიპისა თუ ბმული კავშირების სპეციფიკაციაში წინასწარ განსზღვრულ წესებს.

ჩაწერის ოპერაციების მართვის ჩამოთვლილი მიდგომები შეიცავს გარკვეულ ნაკლოვანებებს. მაგალითად, გარდა იმისა, რომ კლიენტი ხშირ შემთხვევაში დამოკიდებული ხდება გარე ინფორმაციაზე რაც დაუშვებელია ჰიპერმედია სერვისებში. ხშირ შემთხვევაში კი, მსგავსი მიდგომა იწვევს კლიენტის რეალიზაციის პირდაპირ და მჭიდრო დამოკიდებულებას კონკრეტული სერვისისთვის დამახასიათებელ სპეციფიურ დეტალებზე.

2.1. HTTP მეთოდები და მათი სემანტიკა

მომდევნო ქვეთავებში მოკლედ არის დახასიათებული ჩაწერის ოპერაციათა განხორციელებისთვის განკუთვნილი HTTP მეთოდები და მათ გამოყენებასთან დაკავშირებული სპეციფიური დეტალები.

2.1.1. POST მეთოდი

POST მეთოდი, არის ზოგადი დანიშნულების მეთოდი, რომელიც გამოიყენება საწყისი სერვერისთვის მონაცემების გადასაცემად ამ მონაცემების მომდევნო დამუშავებისთვის[3]. მონაცემების დასამუშავებელი(მაგალითად, შექმნა, რედაქტირება) ოპერაციის შერჩევა დამოკიდებულია სხვადასხვა ფაქტორზე და კონკრეტულ შემთხვევაში გადაწყვეტილება მიიღება სერვერის მიერ.

მიუხედავად იმისა, რომ ოპერაციის შედეგად დაბრუნებული პასუხიდან შესაძლებელია გადაცემული მონაცემების საფუძველზე შესრულებული ოპერაციის დადგენა(მაგალითად, HTTP პროტოკოლის სტატუსის კოდის მიხედვით 201 Created), არ არსებობს რაიმე შეთანხმებული საშუალება, რომლის მეშვეობითაც წინასწარ განვსაზღვრავთ მეთოდის დანიშნულებას კონკრეტული შემთხვევისთვის. აღნიშნული შესაძლებლობის არარსებობა განსაკუთრებით მნიშვნელოვანია POST მეთოდის შემთხვევაში, რაც გამომდინარეობს მისი ზოგადი დანიშნულებიდან და შეუძლებელია იგი ცალსახად დავახასიათოთ როგორც რესურსის შექმნის, მოდიფიცირებისა ან სხვა კონკრეტული ოპერაციის განსახორციელებლად განკუთვნილი მეთოდი.

POST მეთოდის სპეციფიკიდან გამომდინარე მის მიერ შესაძლო შესასრულებელი ოპერაციები შესაძლებელია დაფიქსირდეს: შექმნა/ჩამატება - POST_a და შესრულება - POST_p. POST_a შემთხვევაში მეთოდის ფუნქცია დადგენადია მხოლოდ იმ შემთხვევაში თუ მოთხოვნის პასუხად მიღებული შეტყობინება შეიცავს 201Created სტატუსის კოდს და შესაძლებელია ვთქვათ, რომ ამ შემთხვევაში მეთოდის ფუნქცია ხილვადია ნაწილობრივ. POST_p შემთხვევაში მეთოდის ფუნქცია არის სრულიად უხილავი, რამდენადაც არცერთი სხვა წარმატებით განხორციელებული ოპერაციის აღმნიშვნელი 2xx სტატუსის არ მოიცავს კონკრეტულად განხორციელებული ოპერაციის აღმნიშვნელ ინფორმაციას.

განხილული მეთოდის ზოგადი დანიშნულებიდან გამომდინარე თავისთავად იბადება დამატებითი, მეთოდის ფუნქციის განმსაზღვრელი ელემენტის შემოღების აუცილებლობა რომელზე დაყრდნობითაც შესაძლებელი გახდება კლიენტი აპლიკაციის წინასწარი ინფორმირება მეთოდის კონკრეტული მიზნით გამოყენების შესახებ.

2.1.2. PUT მეთოდი

PUT[4] მეთოდი გამოიყენება არსებული რესურსის მდომარეობის, ახალი მდომარეობით სრულად ჩანაცვლების მიზნით. გარდა ამისა, მეთოდის გამოიყენება შესაძლებელია ახალი რესურსის შექმნისთვის თუ წინასწარ იქნება ცნობილი რესურსის იდენტიფიკატორი(URI).

ისევე როგორც, POST მეთოდის შემთხვევაში, არც ამ შემთხვევაში არსებობს წინასწარ შეთანხმებული საშუალება, რომელიც ცალსახად განსაზღვრავს PUT მეთოდის ფუნქციას(ჩანაცვლება, შექმნა) კონკრეტულ შემთხვევაში. აღნიშნული საშუალების არ არსებობის გამო, როგორც წესი ამ მეთოდს გამოიყენებენ მხოლოდ რესურსის მდგომარეობის ახლით ჩანაცვლების ოპერაციების განსახორციელებლად. გარდა ამისა, PUT მეთოდის გამოყენება ახალი რესურსის შექმნის მიზნით ბუნდოვანია რამდენადაც:

- არ არის ცხადი თუ რა საშუალებით უნდა განახორციელოს სერვისმა კლიენტის ინფორმირება და მისთვის ახალი რესურსის მისამართის გადაცემა;
- არ არის ცნობილი თუ რა შემთხვევაში უნდა მოხდეს ახალი რესურსის იდენტიფიკატორის ფორმირება და კლიენტის ინფორმირება.

მიუხედავად ამისა, POST მეთოდისგან განსხვავებით PUT მეთოდის ფუნქცია ხილვადია, რადგან, იმ შემთხვევაში თუ წარმატებით განხორციელებული PUT მოთხოვნის შედეგად მიღებული პასუხი არ შეიცავს სტატუსის კოდს 201 Created, იგულისხმება, რომ განხორციელდა ჩანაცვლების ოპერაცია. ხოლო იმ შემთხვევაშიც კი, თუ მოთხოვნის შესრულების შედეგი დასრულდა ახალი რესურსის შექმნით(რაც დადგენილია პასუხში 201 Created სტატუსის კოდის მეშვეობით), ყოველი მომდევნო მოთხოვნა ამ რესურსის მიმართებაში აუცილებლად დასრულდება ჩანაცვლების ოპერაციით.

3. ჩაწერის ოპერაციათა კოორდინაცია

ჩაწერის ოპერაციათა ეფექტურად სამართავად, მიზანშეწონილად მიგვაჩნია ინფორმაციის შეტანის ფორმის ცნების განსაზღვრა, და ასევე ინფორმაციის შეტანის ფორმისა და მონაცემების წარმოდგენისთვის გამოყენებული მედია ტიპების ერთმანეთისგან სრული გამოიჯვანა. აღნიშნული აუცილებლობა განპირობებულია რამდენიმე ძირითადი მიზეზით:

- მნიშვნელოვანია უკვე არსებული ფართოდ გავრცელებული მედია ტიპების გამოყენების შესაძლებლობის შენარჩუნება;
- უკვე არსებული ფართოდ გავრცელებული მედია ტიპების საფუძველზე ახალი მედია ტიპების შექმნის მიზანშეწონილობა;
- შეუძლებელია უკვე არსებულ თუ მომავალში შექმნილ მედია ტიპში ჩაწერის ოპერაციებისთვის აუცილებელი სემანტიკის მქონე ელემენტების ჩართვა;
- აღნიშნული გამოიჯვანით შესაძლებელს გავხდით ერთხელ შექმნილი(ან შერჩეული) მონაცემების შეტანის აღსაწერი მედია ტიპის გამოყენებას არსებულ მედია ტიპებთან ერთად.

ინფორმაციის სისრულისთვის, მნიშვნელოვანია განვსაზღვროთ ის შესაძლო ელემენტები რომლებიც აუცილებელია ჩაწერის ოპერაციების სამართავად. ჩაწერის ოპერაციის განსახორციელებლად აგენტს(კლიენტს) ესაჭიროება შემდეგი დეტალების ცოდნა:

- მონაცემთა ელემენტები: ამ შემთხვევაში იგულისხმება მონაცემთა ელემენტებისა და ელემენტების ისეთი ატრიბუტების აღწერა როგორც არის: სახელი, მონაცემის ტიპი, შესაძლო დასაშვები მნიშვნელობები, სავალდებულოა თუ არა ელემენტი და ა.შ. აღნიშნულ ინფორმაციაზე დაყრდნობით, შესაძლებელია არა მხოლოდ მონაცემების შემცველი გზავნილის აგება, არამედ, აგენტის მიერ მონაცემების ვალიდაცია რაც კიდევ ერთი დამატებითი ფაქტორია სერვერთან ურთიერთქმედების შესამცირებლად;
- გასაგზავნი მონაცემების კოდირების ფორმატი: მაგ. application/x-www-form-urlencoded ან სხვა შესაძლო ან სასურველი ფორმატი;

- მოთხოვნის გასაგზავნი ვებ მისამართი: აღნიშნული დეტალის ცხადად მითითებით სერვისის უჩნდება კლიენტის კოორდინაციის დამატებითი შესაძლებლობა, ხოლო, კლიენტი თავისუფლდება ყოველგვარი გარე ინფორმაციისა ან გარკვეული დასკვნების კეთების აუცილებლობისგან;

- მოთხოვნის მეთოდი(POST, PUT): როგორც წინა პუნქტში, ამ შემთხვევაშიც მეთოდისა ან მეთოდების ცხადად მითითების შესაძლებლობის არსებობის შემთხვევაში, სერვისიც და კლიენტიც სრულად თავისუფლდება ყოველგვარი წინასწარი შეთანხმებებით გამოწვეული შეზღუდვისგან;

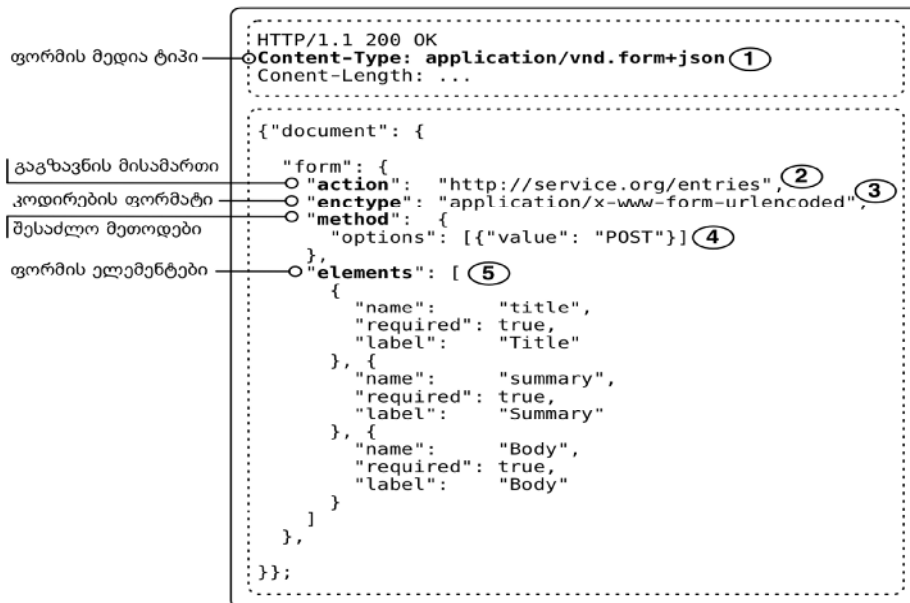
- შერჩეული მოთხოვნის მეთოდის მიერ შესასრულებელი ოპერაციის წინასწარ, უშუალოდ ოპერაციის განხორციელებამდე დადგენა.

მოცემული მომენტისთვის არსებული მედია ტიპებიდან, ჩამოთვლილი შესაძლებლობების ყველაზე სრულყოფილი მხარდაჭერა გააჩნია HTML5 ფორმატს, თუმცა, ჩაწერის ოპერაციების შესასრულებლად გამოყენებადი მოთხოვნის მეთოდების არჩევანი შეზღუდულია მხოლოდ POST მეთოდით და რაც დაუშვებელს ხდის PUT და PATCH მეთოდების გამოყენებას.

ჩამოთვლილი შესაძლებლობების განხორციელებით, და იმის გათვალისწინებით, რომ HTML დოკუმენტის სტრუქტურა ხშირ შემთხვევაში არ არის მისაღები ე.წ. მანქანისა და მანქანათაშორის ურთიერთქმედებისთვის, შესაძლებელია ავგოთ ისეთი მედია ტიპი(ან გამოვიყენოთ არსებული თუ მას გააჩნია სასურველი შესაძლებლობები) რომელიც:

- აღჭურვილი იქნება ყველა აღნიშნული თვისების აღწერის შესაძლებლობით;
- HTML ფორმატისგან განსხვავებით არ იქნება შეზღუდული მხოლოდ POST მეთოდის გამოყენების შესაძლებლობით;
- იქნება ადვილად აღსაქმელი სხვა კლიენტებისთვის.

აღნიშნული მიზნის განსახორციელებლად არჩევანი შევჩერეთ JSON ფორმატზე და შევიმუშავეთ ჰიპოტეტიური ინფორმაციის შეტანის ფორმა რომელიც ნაჩვენებია 1-ელ ნახაზე.



ნახ.1.

მონაცემთა შეტანის ნაჩვენები ფორმის ნიმუში შეიცავს კლიენტის კოორდინაციისთვის საჭირო ინფორმაციას სრულყოფილად. მედია ტიპი(1), მოთხოვნის გაგზავნის მისამართი (2), გასაგზავნი მონაცემების კოდირების ფორმატი (3), მოთხოვნის გასაგზავნად საჭირო მეთოდი (4) და გასაგზავნი მონაცემების სტრუქტურა (5). იმ შემთხვევაში თუ კლიენტისთვის ცნობილია მედია

ტიპის დამუშავების მოდელი (1) მას მარტივად შეუძლია შეტანის ფორმაში აღწერილ დეტალებზე დაყრდნობით მოთხოვნის გზავნილის ფორმირება და სერვერისთვის მონაცემების გადაგზავნა. საინტერესოა რომ, ასეთ შემთხვევაში კლიენტისთვის მნიშვნელოვანია მხოლოდ მედია ტიპების დამუშავების მოდელების წინასწარი ცოდნა. ამ კონკრეტულ შემთხვევაში გამოყენებულია ორი მედია ტიპი:

- application/vnd.form+json რომლის მეშვეობითაც აღწერილია უშუალოდ ინფორმაციის შეტანის ფორმა და
- application/x-www-form-urlencoded რომელიც დღესდღეობით არის გადასაცემ მონაცემთა კოდირების სტანდარტული ფორმატი.

ყოველივე აღნიშნულის გათვალისწინებით მივალწევთ რამდენიმე საგულისხმო მიზანს:

- ჩაწერის ოპერაციის(ამ შემთხვევაში რესურსის შექმნის) განსახორციელებლად აუცილებელი, ყველა სამართავი ელემენტის წარმოდგენა სრულად არის დელეგირებული ინფორმაციის შეტანის ფორმის სტრუქტურის აღსაწერად გამოყენებული მედია ტიპისთვის;
- ინფორმაციის შეტანის ფორმების მართვა სრულად არის გამიჯნული სხვა, მხოლოდ წაკითხვისთვის განკუთვნილი მედია ტიპებისგან.

4. ბმული კავშირების განსაზღვრა

წინამორბედ თავებში განხილული ინფორმაცია სრულყოფილად აღწერს ჩაწერის ოპერაციებისა და მათი კოორდინაციის პრობლემატიკას. ამ ინფორმაციის გათვალისწინებით შესაძლებელია ჩამოვყალიბოთ კავშირი მონაცემების წაკითხვისთვის განკუთვნილ რეპრეზენტაციებსა და ჩაწერის ოპერაციებისთვის განკუთვნილი მონაცემების შეტანის ფორმების რეპრეზენტაციებს შორის.

გამომდინარე იქედან, რომ მოვხდინეთ ჩაწერის ოპერაციებთან დაკავშირებული დეტალების სრული გამიჯვნა, აღნიშნული კავშირების განსაზღვრად საკმარისი იქნება მხოლოდ შესაბამისი სემანტიკის მქონე ბმული კავშირების შემოღება.

ანალიზის შედეგად გამოვყავით ჩაწერის ოპერაციათა ორი ზოგადი ტიპი - შექმნა და განაზღვრა. როგორც ერთ, ასევე მეორე შემთხვევაში უმნიშვნელოა ჩაწერის ოპერაციების განსახორციელებლად აუცილებელი დეტალები, რამდენადაც მათი დელეგირება სრულად მოვხდინეთ მონაცემთა შეტანის ფორმებში. რაც საკმარისი პირობაა შესაბამისი ბმული კავშირების განსაზღვრად, რამდენადაც ეს ბმული კავშირები არ საჭიროებს დამატებით ინფორმაციას, გარდა მკაფიოდ განსაზღვრული სემანტიკისა.

აღნიშნული მიზნით შევქმენით შესაბამისი სპეციფიკაცია, რომელიც განსაზღვრავს ორ ბმულ კავშირს: **create-form** და **edit-form**¹. ბმულ კავშირებს გააჩნიათ ზოგადი, თუმცა ამავედროულად მკაფიო სემანტიკა და სპეციფიკაციის მიხედვით მათი გამოყენება უნდა მოხდეს მაშინ, როდესაც საჭიროა კლიენტისთვის მონაცემების შეტანის(როგორც შექმნის ასევე მოდიფიცირების) შესახებ დამატებითი ინფორმაციის მიწოდება. აღსანიშნავია, რომ, არცერთი ბმული კავშირი არ არის დამოკიდებული რომელიმე კონკრეტული მედია ტიპის სპეციფიკაციისა თუ სერვისზე. მათი გამოყენება შესაძლებელია ნებისმიერ უკვე არსებულ, თუ მომავლ მედია ტიპთან ერთად და მათი გამდიდრება აღნიშნული შესაძლებლობებით. აღნიშნული ბმული კავშირებიდან:

- **create-form** ბმული კავშირი განკუთვნილია ისეთი ბმულების აღსაწერად, რომლებიც მიუთითება მონაცემთა შეტანის ფორმებს. მათი განმისამართებით აგენტი მიიღებს

¹ მოცემული მომენტისთვის აღნიშნული ბმული კავშირები რეგისტრირებულია IETF - ის მიერ როგორც ინფორმაციული სტანდარტი. RFC6861 [11]

შექმნისობერაციასთან დაკავშირებულ სრულ წარმოდგენას(მონაცემთა სტრუქტურა, მოთხოვნის გაგზავნის მისამართი, მოთხოვნის მეთოდი(ან მეთოდები));

- **update-form** ბმული კავშირი განკუთვნილია ისეთი ბმულების აღსაწერად, რომლებიც მიუთითებს მონაცემთა შეტანის ფორმებს და რომელთა განმისამართებით კლიენტი მიიღებს წინასწარ შევსებულ შეტანის ფორმას, რომელიც ასევე შეიცავს სხვა აუცილებელ სამართავ ელემენტებს.

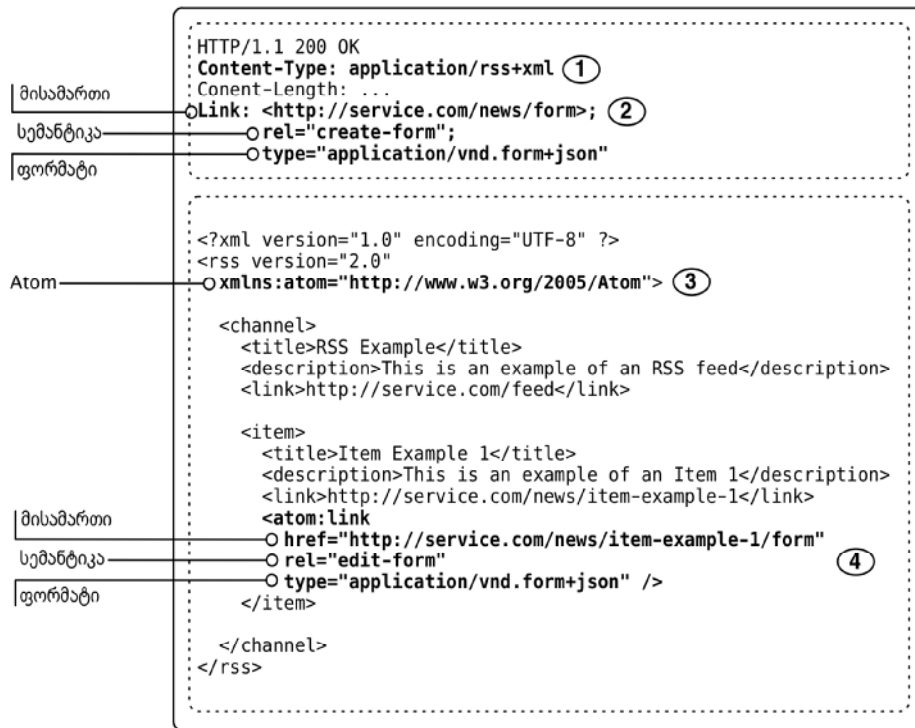
მე-2 ნახაზზე ნაჩვენებია დოკუმენტი შეიცავს რამდენიმე მნიშვნელოვან დეტალს, როგორცაა:

- შენარჩუნებულია საბაზისო application/rss+xml მედია ტიპი (1) რაც უზრუნველყოფს უკვე არსებული კლიენტების მხარდაჭერას;

- HTTP პროტოკოლის Link თავსართის მეშვეობით მიუთითებულია ინფორმაციის შეტანის ფორმის ვებ მისამართი რომელიც ანოტირებულია შესაბამისი ბმული კავშირით (2).

- გამოდინარე იქედან, რომ RSS მედია ფორმატს არ გააჩნია ანოტირებული ბმულების მხარდაჭერა, იგი განვრცობილია Atom სპეციფიკაციის მეშვეობით (3) ისე როგორც რეკომენდებულია უშუალოდ აღნიშნული ფორმატის სპეციფიკაციისა და ზოგადად XML სპეციფიკაციის მიერ;

- დოკუმენტში ჩართულ კონკრეტულ ჩანაწერს თანდართული აქვს Atom ბმული რომელშიც მიუთითებულია აღნიშნული ჩანაწერის რედაქტირებისთვის განკუთვნილი შეტანის ფორმის ვებ მისამართი რომელიც ანოტირებულია შესაბამისი ბმული კავშირით (4). იმ შემთხვევაში, თუ კლიენტისთვის ცნობილია როგორც Atom ბმულის სტრუქტურა, ასევე მის მეშვეობით მიუთითებული ბმული კავშირის სემანტიკა და მედია ტიპის დამუშავების მოდელი, ჩანაწერის რედაქტირების აუცილებლობის შემთხვევაში კლიენტს შეუძლია მიიღოს საკმარისი ინფორმაცია მიუთითებული რესურსის განმისამართებით.



ნახ.2. RSS ფორმატის გადმოდგენა Atom ბმულითა და ბმული კავშირებით

5. დასკვნა

ნაჩვენები მიდგომით შესაძლებელია როგორც უკვე არსებული კლიენტების შენარჩუნება მათი მოდიფიცირების გარეშე, ასევე ახალი კლიენტების შექმნა რომლებსაც გარდა წაკითხვისა, საჭიროების შემთხვევაში შეძლებენ ჩაწერის ოპერაციების განხორციელებას. საინტერესოა, რომ გარდა სხვა დეტალებისა მიდგომა უზრუნველყოფს წაკითხვისთვის განკუთვნილი ფორმატების განტვირთვის რამდენიმე თვალსაზრისით:

- წაკითხვისთვის განკუთვნილ რეპრეზენტაციებში საკმარისია მხოლოდ შესაბამისი ბმულებისა და ბმული კავშირების მითითება;
- რიგ შემთხვევებში ბმულებისა და ბმული კავშირების მითითება შესაძლებელია HTTP პროტოკოლის Link სათაურის მეშვეობით;
- ჩაწერისთვის აუცილებელი მონაცემთა მოდელისა და სხვა სამართავი ფაქტორების უზრუნველყოფა სრულიად არის დელეგირებული შესაბამისი მედია ტიპებისთვის.
- ჩაწერისთვის განკუთვნილი მედია ტიპის შეცვლის შემთხვევაში, წაკითხვისთვის განკუთვნილი ფორმატი რჩება უცვლელი;

ლიტერატურა:

1. Fielding, Roy Thomas. Architectural Styles and the Design of Network-based Software Architectures. Doctoral dissertation, University of California, Irvine, 2000.
2. Hypertext Transfer Protocol - HTTP/1.1
<<http://www.w3.org/Protocols/rfc2616/rfc2616.html>>
3. Method Definitions, POST <<http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html#sec9.5>>
4. Method Definitions, PUT <<http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html#sec9.6>>
5. PATCH Method for HTTP. RFC 5789 <<http://tools.ietf.org/html/rfc5789>>
6. Uniform Resource Identifiers (URI) <<http://www.ietf.org/rfc/rfc2396.txt>>
7. Web Linking <<http://tools.ietf.org/html/rfc5988>>
8. Link Relations <<http://www.iana.org/assignments/link-relations/link-relations.xml>>
9. Create, read, update and delete
<http://en.wikipedia.org/wiki/Create,_read,_update_and_delete>

10. Jim Webber, Savas Parastatidis and Ian Robinson. REST in Practice: Hypermedia and Systems Architecture. O'Reilly Media; 1 edition (Sep 24, 2010)

11. The "create-form" and "edit-form" Link Relations. RFC6861
<<http://tools.ietf.org/html/rfc6861>>

12. The Atom Publishing Protocol <<https://tools.ietf.org/html/rfc5023>>

COORDINATING WRITE OPERATIONS WITH SEMANTIC LINKS

Ioseb Dzmanashvili, Zurab Bosikashvili

Georgian Technical University

Resume

Semantic links play central role in machine to machine interaction oriented hypertext APIs. This paper describes the process of defining new link relation types, outlines benefits and possible ways of extending semantics of already existing media formats.

КООРДИНАЦИЯ ОПЕРАЦИИ ЗАПИСИ С ПОМОЩЬЮ СЕМАНТИЧЕСКИХ ГИПЕРССЫЛОК

Дзманашвили И., Босикашвили З.

Грузинский Технический Университет

Резюме

Семантические гиперссылки играют центральную роль в гипертекстовых сервисах ориентированных на межмашинное взаимодействие. В статье рассмотрен процесс создания новых семантических гипертекстовых соотношении, и пути улучшения семантики уже установленных медиа типов.