**REST**

Το REST (Representational State Transfer) αποτελεί ένα σύνολο από αρχές σχεδίασης μιας δικτυακής υπηρεσίας που επικεντρώνει στους πόρους ενός συστήματος. Η μεταβολή της κατάστασης των πόρων του συστήματος περιγράφεται και μεταφέρεται στο σύστημα μέσω του πρωτοκόλλου HTTP από διάφορους clients (ανεξαρτήτως της γλώσσας στην οποία έχουν υλοποιηθεί). Το το REST πρωτοεμφανίστηκε το 2000 από τον Roy Fielding στην ακαδημαϊκή του διατριβή με τίτλο «Architectural Styles and the Design of Network-based Software Architectures». Η διατριβή του Fielding εξήγησε τις αρχές REST που ήταν γνωστές ως "πρότυπο αντικειμένου HTTP" που ξεκίνησε το 1994 και χρησιμοποιήθηκαν για το σχεδιασμό των προτύπων HTTP.

Τα βασικά χαρακτηριστικά του REST είναι τα εξής:

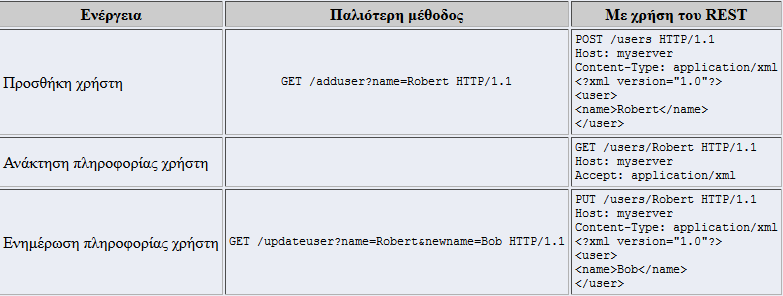
**1. Αποκλειστική χρήση HTTP αιτημάτων/μεθόδων για την επικοινωνία του χρήστη με τον παροχέα της δικτυακής υπηρεσίας.**

Η βασική αρχή σχεδίασης του REST είναι η ένα-προς-ένα αντιστοίχιση μεταξύ λειτουργιών CRUD (create, read, update, delete) και HTTP μεθόδων. Σύμφωνα με αυτή την αντιστοίχιση:

* Για τη δημιουργία ενός πόρου στον server, χρησιμοπούμε την μέθοδο POST.
* Για την ανάσυρση ενός πόρου, χρησιμοποιούμε την GET.
* Για την αλλαγή της κατάστασης ενός πόρου ή την ενημέρωσή του, χρησιμοποιούμε την PUT.
* Για την απομάκρυνση ή διαγραφή ενός πόρου, χρησιμοπουούμε την DELETE.

Με βάση το REST το URI δεν χρησιμοποιείται πια για την περιγραφή της ενέργειας που θέλουμε να εκτελέσουμε αλλά μόνο τον εντοπισμό του πόρου επί του οποίου θα ασκηθεί η ενέργεια, και τα δεδομένα δεν μεταφέρονται ως παράμετροι στο URI ενός GET αιτήματος αλλά ως XML ή JSON-formatted δεδομένα στο περιεχομενα μιας POST ή PUT μεθόδου. Mε άλλα λόγια σε μια υπηρεσία REST, ένα URI εκφράζει ένα αντικείμενο στο οποίο παρέχει πρόσβαση η υπηρεσία μέσω ενός HTTP αιτήματος. Το είδος του αιτήματος καθορίζει την ενέργεια που θέλουμε να εφαρμόσουμε στο αντικείμενο αυτό και το περιεχόμενο του αιτήματος περιέχει διάφορες εξειδικεύσεις τις ενέργειας.

Παραδείγματα:



**2. Stateless**

Η ουσία της «έλλειψης κατάστασης» είναι ότι οποιαδήποτε κλήση σε μια υπηρεσία REST δε θα πρέπει να αναφέρεται σε άλλη προγενέστερη κλήση. Όλες οι κλήσεις πρέπει να είναι ανεξάρτητες. Ο server δε θα πρέπει να γνωρίζει το έγινε με κάποια προηγούμενη κλήση τη στιγμή που επεξεργάζεται την τρέχουσα κλήση. Σκεφτείται για παράδειγμα μια δικτυακή υπηρεσία στην οποία ο πελάτης θα πρέπει πρώτα να κάνει πρώτα μια κλήση για login στην υπηρεσία και μέτά μια άλλη κλήση για ανάκτηση της πληροφορίας που επιθυμεί. Αυτή η υπηρεσία δεν είναι stateless γιατί κατά την επεξεργασία της δεύτερης κλήσης η υπηρεσία θα πρέπει να θυμάται ότι αυτός ο χρήστης έχει ήδη συνδεθεί μέσω μιας προηγούμενης κλήσης. Η κατάσταση «συνδεδεμένος» διατηρείται από την υπηρεσία και ο πελάτης δεν χρειάζεται να παρουσιάσει τα πιστοποιητικά του σε περίπτωση μιας δεύτερης ή τρίτης κλήσης της υπηρεσίας. Μια τέτοια διατήριση καταστάσεων απαγορεύεται σε δικτυακές υπηρεσίες τύπου REST.

Μία υπηρεσία ή χρήστης του REST πρέπει να περιλαμβάνει εντός του HTTP αιτήματος (header και body) όλη την πληροφορία (action, parameters, context,...) που είναι απαραίτητη από τον server για την παραγωγή μιας απόκρισης στο αίτημα αυτό. Τα αιτήματα πρέπει να είναι ολοκληρωμένα και αυτόνομα έτσι ώστε κατά την επεξεργασία τους από τον server να μην απαιτούν την ανάσυρση κάποιας άλλης πληροφορίας η οποία να επηρεάζει την απόκριση στο ερώτημα. Αυτή η «έλλειψη κατάσταση» σε μια υπηρεσία REST απαλλάσει τον server από την ανάγκη συγχρονισμού των δεδομένων μιας συνόδου με μια εξωτερική εφαρμογή. Ακόμα περισσότερο επιτρέπει στις υπηρεσίες REST να προσαρμόζονται σε περιπτώσεις όπου απαιτούνται υψηλές επιδόσεις χρησιμοποιώντας clusters of servers με διαμοιρασμό φορτίου (load-balancing) και δυνατότητες για failover, proxies και άλλες τοπολογίες οι οποίες θα προωθούν τα αιτήματα από τον ένα server στον άλλο με σκοπό την ελάττωση του συνολικού χρόνου απόκρισης στην κλήση μιας δικτυακής υπηρεσίας.

**3. Απεικόνιση της δομής των καταλόγων σαν URIs.**

Ο σχηματισμός των URI μιας υπηρεσίας REST πρέπει να είναι προβλέψιμος και ευνόητος για τον χρήστη της υπηρεσίας και να απαιτείται από λίγη έως καθόλου τεκμηρίωση ή εξήγηση σχετικά με τον πόρο προς τον οποίο δείχνει κάθε URI. Ένας τρόπος να το πετύχουμε αυτό είναι να ορίσουμε URI παρόμοια με τον τρόπο δόμησης των καταλόγων σε ένα σύστημα αρχείων. Για παράδειγμα η δομή ενός URI που δείχνει σε ένα συγκεκριμένο θέμα συζήτησης μέσα σε ένα φόρουμ συζητήσεων θα μπορούσε να είναι:

http://www.myservice.org/discussion/topics/{topic}

ή

http://www.myservice.org/discussion/{year}/{day}/{month}/{topic}

Μερικά σημεία τα οποία πρέπει να έχουμε υπόψιν όταν σχεδιάζουμε την δομή των URI σε μια υπηρεσία REST είναι και οι παρακάτω:

* Αποκρύψτε τις επεκτάσεις των αρχείων που αφορούν την τεχνολογία προγραμματισμού στην πλευρά του server (.jsp, .php, .asp), αν υπάρχουν κάποιες φυσικά, έτσι ώστε αν θέλετε να αλλάξετε κάποια στιγμή το script στο οποίο δείχνει το URI να μην χρειάζεται αν αλλάξετε και το URI.
* Χρησιμοποιείται πάντα μικρουύς και όχι κεφαλαίους χαρακτήρες.
* Αντικαταστήστε το κενά διαστήματα (στα URI) με παύλες ή κάτω παύλες.
* Αποφύγετε τα query strings όσο το δυνατόν.
* Αντί να χρησιμοποιείται τον κωδικό "404 Not Found " όταν το URI στο οποίο στάλθηκε το αίτημα είναι μόνο τμήμα ενός αποδεκτού URI, χρησιμοποιήστε μια default σελίδα σαν απόκριση.

Τα URI θα πρέπει να είναι στατικά, έτσι ώστε όταν οι πόροι της υπηρεσίας υφίστανται αλλαγές (προσθήκες, ενημερώσεις, διαγραφές, κ.λ.π) ή αλλάζει η υλοποίηση της υπηρεσίας, ο σύνδεσμος προς έναν πόρο του συστήματος να παραμένει ίδιος. Αυτό επιτρέπει τη δημιουργία σελιδοδεικτών.

**4. Μεταφορά δεδομένα, και προς τις δύο διευθύνσεις, μέσω XML, JSON ή και των δύο.**

Για να δώσουμε στον client την δυνατότητα να επιλέξει τη μορφή των περιεχομένων της απόκρισης του server στην πιο κατάλληλη γι αυτόν,δεν έχουμε παρά να φτιάξουμε την υπηρεσία μας έτσι ώστε να κάνει χρήση του ενσωματομένου HTTP Accept header όπου η τιμή του θα είναι ένας τύπος MIME. Οι πιο συνηθισμένοι τύποι μορφοποίησης δεδομένων στις REST υπηρεσίες είναι JSON (application/json), XML (application/xml) και XHTML(application/xhtml+xml).

**Not exaclty REST-full:** Πολλές υπηρεσίες σήμερα που διαφημίζονται ως REST δεν πληρούν ακριβώς τις προδιαγραφές μιας υπηρεσίας REST. Για παράδειγμα χρησιμοποιούν ρήματα μέσα στα URI για να καθορίσουν την επιθυμητή ενέργεια π.χ /users?method=createUserAccount or even /users/create . Επίσης, πολλοί υπερφορτώνουν τις μεθόδους GET και POST με σκοπό να υλοποιήσουν αιτήματα PUT και DELETE. Αυτό γίνεται χρησιμοποιώντας ένα custom HTTP header, π.χ το X-HTTP-Method-Override header:

POST /bookmarkservice/skonnard/bookmarks/123 HTTP/1.1   
X-HTTP-Method-Override: DELETE

**Τσιόγκας Ηλίας 3778**

**ΒΙΒΛΙΟΓΡΑΦΙΑ:**

<https://en.wikipedia.org/wiki/Representational_state_transfer>

<https://www.codecademy.com/articles/what-is-rest>

<https://www.service-architecture.com/articles/web-services/representational_state_transfer_rest.html>