**Įvadas**

Šis dokumentas aprašo testavimo planą serverio užsakymo proceso iš vartotojo pusės patikrinimui. Šio plano tikslas yra užtikrinti, kad serverio užsakymas būtų sėkmingas, kai vartotojas bando užsisakyti serverius su įvairiais parametrų rinkiniais. Dokumentas apima testavimo plano apimtį, testavimo tikslus, testavimo strategiją, testavimo rezultatų pateikimą, testavimo aplinką ir testavimo planą.

**Apimtis**

* Testuoti API taško serverio užsakymui veikimą siunčiant užklausas su atsitiktiniais parametrais.
* Testuoti API taško gebėjimus apdoroti klaidingas arba netikėtas užklausas.
* Patikrinti ar atsakymai į užklausas pateikia tinkamus duomenis bei HTTP statuso kodus.

Į testavimo apimtį neįeina užklausų formavimui reikalingų tinkamų duomenų gavimo API taškų funkcionalumo patikrinimas:

* Visų galimų regionų sąrašo gavimas.
* Visų galimų planų sąrašo gavimas.
* Kiekvieno plano galimų OS atvaizdų sąrašo gavimas.

**Testavimo tikslai**

* Įsitikinti, kad API gali apdoroti netikėtas arba netinkamas užklausas nesukeliant sistemos trikdžių.
* Patikrinti iš API gautus rezultatus.
* Registruoti užklausų ir atsakymų duomenis ir suteikti galimybę atlikti rezultatų analizę.

**Testavimo strategija**

Naudojama ,,Juodos dėžės“ testavimo metodika API testavimui, nežinant vidinio API kodo ir veikimo. Testavimas atliekamas automatiškai, sukuriant programą ,,beždžionę“, kuri imituoja vartotoją ir tam tikru laiko intervalu bando užsakyti serverį atsitiktinai parenkant galimus parametrus. Parametrai gaunami kreipiantis į kitus API taškus, kurių veikimo tinkamumas nėra testuojamas, todėl, klaidos atveju, ,,beždžionė“ nusiųs netinkamą užklausą.

**Testavimo rezultatai**

Gauti testavimo rezultatai bus saugomi MySQL duomenų bazėje. Testo rezultatuose bus nurodyta kiekvieno testo būsena, vykdymo laikas, unikalus ID ir išsaugoti užklausos bei gauto atsakymo duomenys. Duomenims saugoti bus sukurtos kelios atskiros lentelės su esminiais laukais, siekiant rezultatus padaryti labiau filtruojamus.

**Testavimo aplinka**

Testavimas atliekamas su realiu produkcijoje veikiančiu API. Programos ,,beždžionės“ rašymui naudojama C# programavimo kalba ir jos naudojamos bibliotekos bei karkasai:

* .NET 7.0
* RestSharp 110.2.0
* Newtonsoft.Json 13.0.3
* Microsoft.NET.Test.Sdk 17.5.0
* Microsoft.Extensions.Configuration.Json 7.0.0
* NUnit 3.13.3
* NUnit3TestAdapter 4.4.2

Parašyta programa yra konsolės tipo ir sukompiliuotas vykdomasis failas gali būti paleidžiamas skirtingose operacinėse sistemose, kuriose įdiegtas tinkamas .NET Runtime.

Rezultatų duomenims saugoti naudojama MySQL duomenų bazės valdymo sistema.

**Testavimo planas**

Testavimo scenarijai:

* Užsakomas serveris su standartiniais parametrais.
* Užsakomas serveris su nestandartiniais tinkamais parametrais.
* Užsakomas serveris su klaidingais parametrais.

Testavimo atvejai:

|  |  |
| --- | --- |
| Įvestis | Numatomas rezultatas |
| Užklausa su klaidingais parametrais | HTTP status code 400 |
| Užklausa su standartiniais parametrais – pirmas regionas, pirmas planas, pirmas OS atvaizdas, nepasirinkti kiti papildomi parametrai | HTTP status code 201, atsakymo JSON objektas su standartiniais parametrais ir užsakytas atitinkamas serveris. |
| Užklausa su atsitiktiniais galimais parametrais. | HTTP status code 201, atsakymo JSON objektas su užpildytais atitinkamais parametrais ir užsakytas tinkamas serveris. |
|  |  |
|  |  |
|  |  |

Testavimo vykdymas – automatinis ,,Beždžionės“ paleidimas kartą per valandą. Programos veikimo principas:

1. Per API taškus surenkami galimi duomenys užklausai formuoti:

* Regionų sąrašas.
* Planų sąrašas.
* Kiekvieno plano OS atvaizdų sąrašas.
* ...

1. Suformuojama užklausa su atsitiktiniais parametrais.
2. Vykdomas kreipimasis į API tašką siunčiant sugeneruotą užklausą.
3. Gaunamas teigiamas arba neigiamas rezultatas.
4. Rezultato duomenys išsaugomi duomenų bazėje.