

Nowa koncepcja nazewnictwa

Ogólna struktura nazw repozytoriów

Ogólne zasady

- w poniższych podpunktach znajduje się opis zasad nazewnictwa repozytoriów w podziale na różne ich typy
- na końcu znajduje się zestawienie słowników
- poniższe segmenty są zesłownikowane
 - rodzaj
 - przeznaczenie
- platforma - należy ograniczyć liczbę występujących nazw platform do absolutnego minimum
- jeżeli należy dodać dodatkowy słownik, prośba o kontakt z zespołem DevOps (skrzynka devops.zlecenia@ergohestia.pl, kontakt na Teams z przedstawicielem zespołu)
- poszczególne segmenty rozdzielane są kropką
- jeżeli dany segment składa się z kilku słów, rozdzielimy je myślnikiem
- nazwy aplikacji powinny być opisowe, ale w rozsądny zakresie
- jeżeli nazwy aplikacji są skrócone, pełną nazwę aplikacji należy zawrzeć w opisie
- w przypadku testów zakładamy możliwość tworzenia kilku repozytoriów, każde z nich na końcu po myślniku powinno

Aplikacje oraz biblioteki przypisane do platformy

[platforma].[nazwa-aplikacji].[rodzaj].[przeznaczenie]

np.:

ihestia.ssoag.frontend.deploy - kod deployujący frontend aplikacji ssoag wchodzącej w skład platformy iHestia
ihestia.ssoag.frontend.src - źródła frontendu aplikacji ssoag wchodzącej w skład platformy iHestia

bos.salesnetwork.fullstack.src - źródła fullstackowej aplikacji salesnetwork wchodzącej w skład platformy BOS
bos.salesnetwork.fullstack.deploy - kod deployujący fullstackową aplikację salesnetwork wchodząą w skład platformy BOS
bos.salesnetwork.fullstack.test{-dowolny} - kod testujący fullstackową aplikację salesnetwork wchodzącą w skład platformy BOS
bos.salesnetwork.fullstack.test{-dowolny} - kod testujący fullstackową aplikację salesnetwork wchodząą w skład platformy BOS
bos.salesnetwork.fullstack.test{-dowolny} - kod testujący fullstackową aplikację salesnetwork wchodząą w skład platformy BOS

devops.zroaks.scripts.deploy - generyczne skrypty devops związane z AKSem ZRO
devops.default.scripts.deploy - generyczne skrypty deployujące devops
devops.tools.scripts.deploy - generyczne skrypty narzędziowe devops

UWAGA! Czasami na frontend czy backend składa się więcej, niż jedno repozytorium, które rozróżnić można jedynie poprzez technologię (np. CRM). W takim wypadku dopuszcza się, żeby w drugim członie występowała technologia, np.:

crm.typescript.frontend.src
crm.angular.frontend.src

Aplikacje, które nie są przypisane do platformy

[nazwa-aplikacji].default.[rodzaj].[przeznaczenie]

np.:

apk.default.fullstack.src
apk.default.lib.src
crm.default.backend.src

Biblioteki współdzielone

common.lib.[technologia].[nazwa-biblioteki]

np.:

common.lib.java.gui

Słowniki

Platforma

ihestia
ocean
bos
devops
service - małe serwisy, współdzielone przez wiele platform

Nazwa aplikacji (przykłady):

clientservice
agentservice
sale
tarrification
pinpoint
produkty
ssoag-oauth

default (jeżeli rozwiązanie nie jest podzielone na obszary)

Rodzaj

backend - backend rozwiązania, które składa się również z części frontend
frontend - frontend rozwiązania, które składa się również z części backend
lib - biblioteka/zestaw bibliotek
api - rozwiązanie, które będzie serwołało API
scripts - skrypty (np. terraform, githubactions, Oracle, SQLServer)
doc - dokumentacja
fullstack - aplikacja, która nie jest podzielona na warstwy frontend i backend, posiadająca obie warstwy w jednym repozytorium
handler - kod obsługujący kolejki

Przeznaczenie

tests - repo z automatami/konfiguracją testów
deploy - repo z definicją CD
secret - repo z danymi wrażliwymi
src - źródła rozwiązania