# Przed L2

REST – Representational State Transfer – styl architektury oprogramowania opierający się o zbiór wcześniej określonych reguł opisujących jak definiowane są zasoby, a także umożliwiających dostęp do nich. Został on zaprezentowany przez Roya Fieldinga w 2000 roku.

API – Application Programming Interface – zestaw reguł definiujący komunikację pomiędzy programami komputerowymi.

Czyli API są to reguły określające jak użytkownik może uzyskać dostęp do zasobów oraz w jakiej postaci je otrzymuje. Natomiast REST to styl architektury definiujący jak zbudowane będzie to API.

HTTP – Hypertext Transfer Protocol – protokół, z którego korzystasz codziennie (lub też jego wersji szyfrowanej – HTTPS) podczas przeglądania stron w sieci. Podczas tworzenia REST API do komunikacji z API wykorzystuje się metody HTTP, których łącznie jest 9. Niemniej jednak do zbudowania podstawowego API pozwalającego na odczyt, zapis, aktualizację i usuwanie danych wystarczą tylko 4 metody – GET, POST, PUT i DELETE.

Cykl pracy z API możemy określić następująco:

Klient preparuje zapytanie w postaci odpowiedniego adresu (endpointa).

Następnie wysyła to zapytanie do interfejsu.

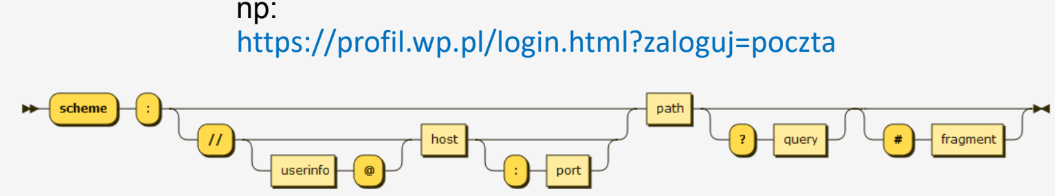
Interfejs zwraca odpowiedź na zapytanie klienta.

Klient otrzymuje odpowiedź na swoje zapytanie.

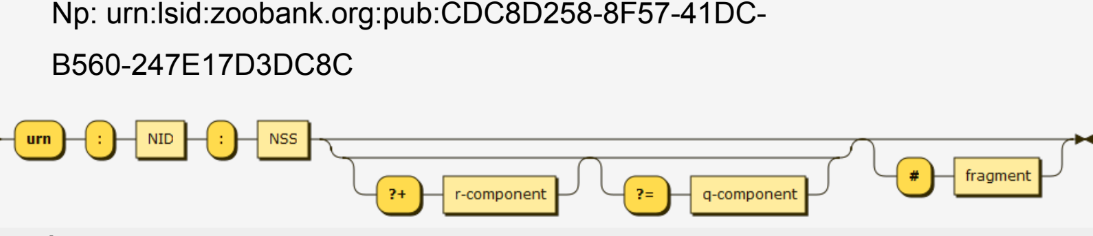
Prezentacja

URI - Uniform Resource Identifier - standard sieciowy definiujący sposób metodę zapisu identyfikatorów umożliwiających łatwą i jednoznaczną identyfikację zasobów.

URL - Unified Resource Locator - Ujednolicony format adresowania zasobów umożliwiający łatwą i jednoznaczną lokalizację zasobu. RodzajURI (najczęściej z URL utoższamiane jest URI)



URN - Unified Resource Name - ujednolicony format nazw zasobów. Składa się z identyfikatora przestrzeni nazw (NID) oraz ciągu znaków specyficznego dla danej przestrzeni (NSS).



HTTP (ang. Hypertext Transfer Protocol) –Za pomocą protokołu HTTP przesyła się żądania udostępnienia dokumentów WWW i informacje o kliknięciu odnośnika oraz informacje z formularzy. Zadaniem stron WWW jest publikowanie informacji – natomiast protokół http właśnie to umożliwia.

-Bezpołączeniowy

- Niezależny od zawartości (media independent)

- Bezstanowy

**DNS rozwiązuje nazwy domen na adresy IP**

ŻĄDANIA/METODY:

- GET – pobierz zasób

- POST – umieść zasób

- HEAD – pobierz nagłówki

- PUT – aktualizuj zasób

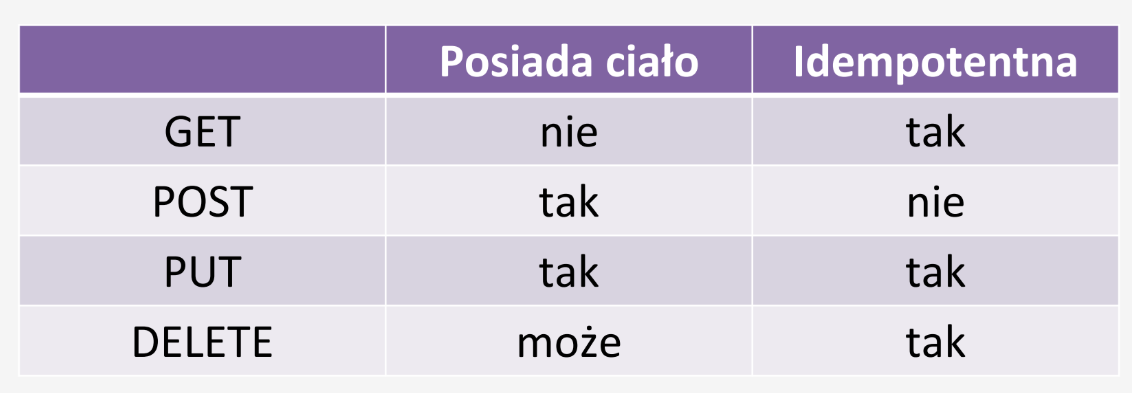
- DELETE – usuń zasób

- CONNECT – utwórz połączenie

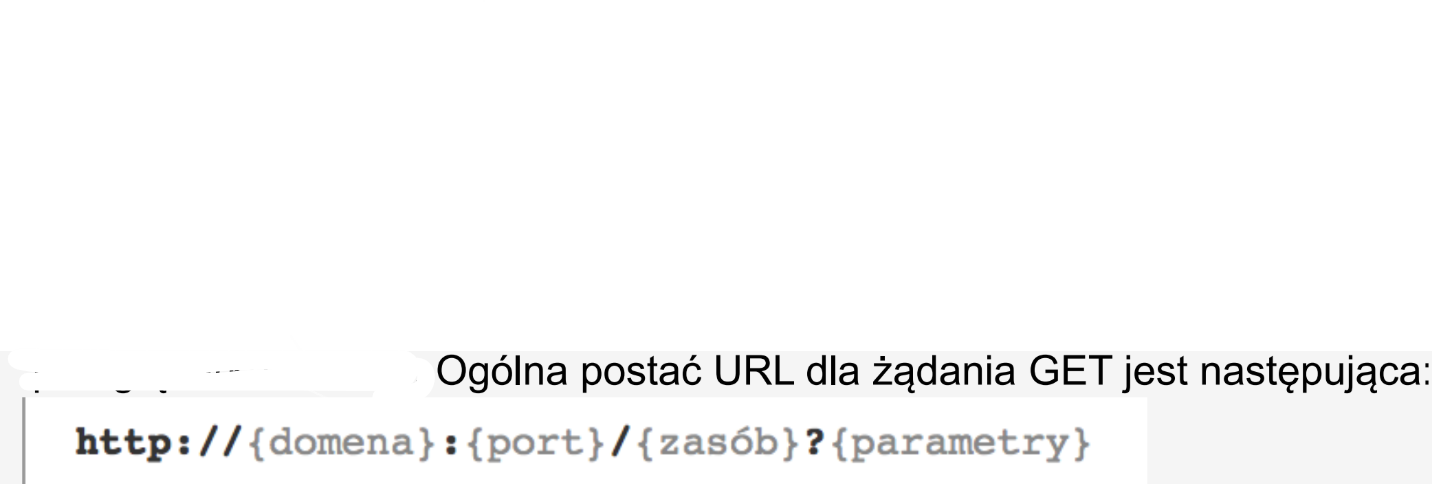
- OPTIONS – jakie rządania obsługuje serwer

- TRACE – test

Idempotentność – właściwość pewnych operacji, która pozwala na ich wielokrotne stosowanie bez zmiany wyniku.



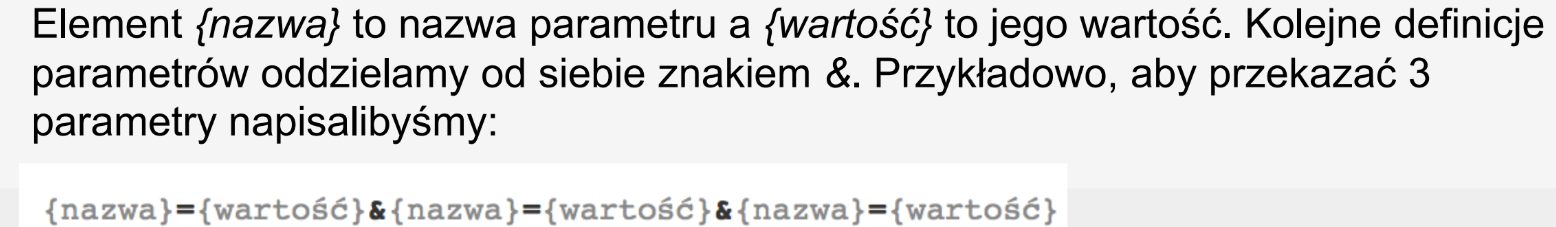


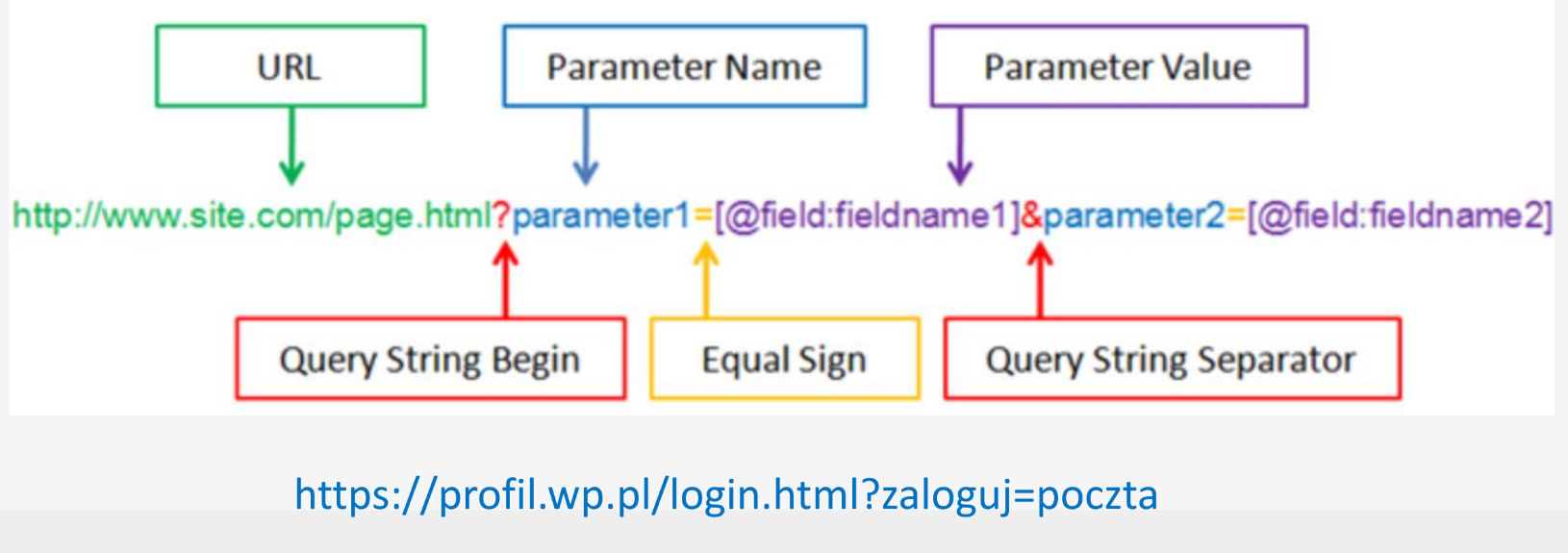


Zarówno {port} jak i {zasób} są opcjonalne. Opcjonalne są także {parametry}, ale jeśli

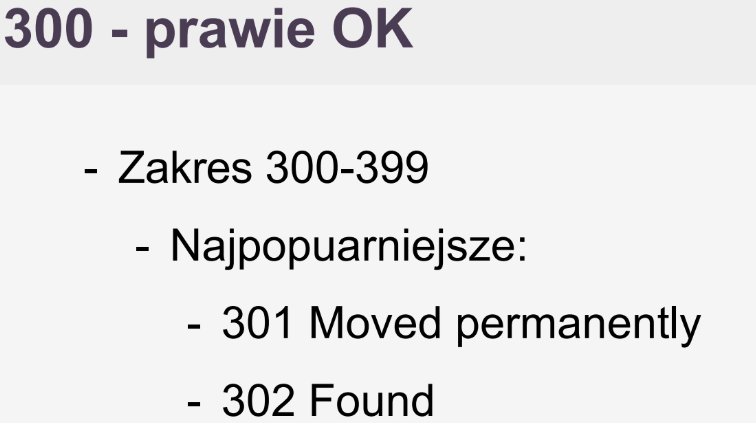
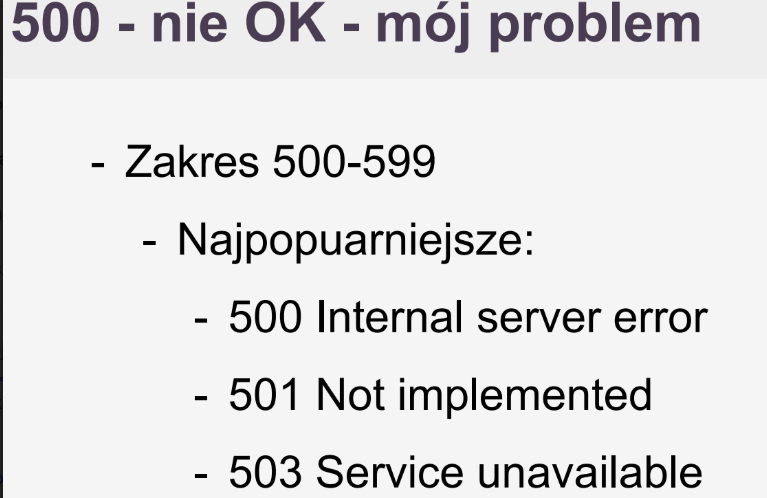
chcemy je przekazać wraz z żądaniem, to doklejamy je do końca URL po znaku

zapytania.





ODPOWIEDZI:

* 
* 
* 
* 

HTTPS (ang. Hypertext Transfer Protocol Secure) – szyfrowana wersja protokołu HTTP. W przeciwieństwie do komunikacji niezaszyfrowanego tekstu w HTTP klient-serwer, HTTPS szyfrował dane przy pomocy protokołu SSL, natomiast obecnie używany jest do tego celu protokół TLS. Zapobiega to przechwytywaniu i zmienianiu przesyłanych danych.

- Zapewnia, że nikt tak łatwo naszej transmisji nie podejrzy

- Najfajniejsze certyfikaty, to takie, które wystawiła zaufana trzecia strona

- Ale bez tych certyfikatów nie podejrzymy sobie komunikacji w

Wiresharku

**REST**

**REpresentational State Transfer**

**1. Odseparowanie interfejsu użytkownika od operacji na serwerze**

klient nie ma wpływu na serwer i odwrotnie

**2. Bezstanowość**

zapytania z kompletem informacji, brak przechowywania stanu sesji po

stronie serwera

**3. Cacheability**

serwer musi jasno określać czy coś jest keszowalne czy nie

**4. Endpointy**

jednoznaczene wskazywanie zasobu do którego klient się odwołuje

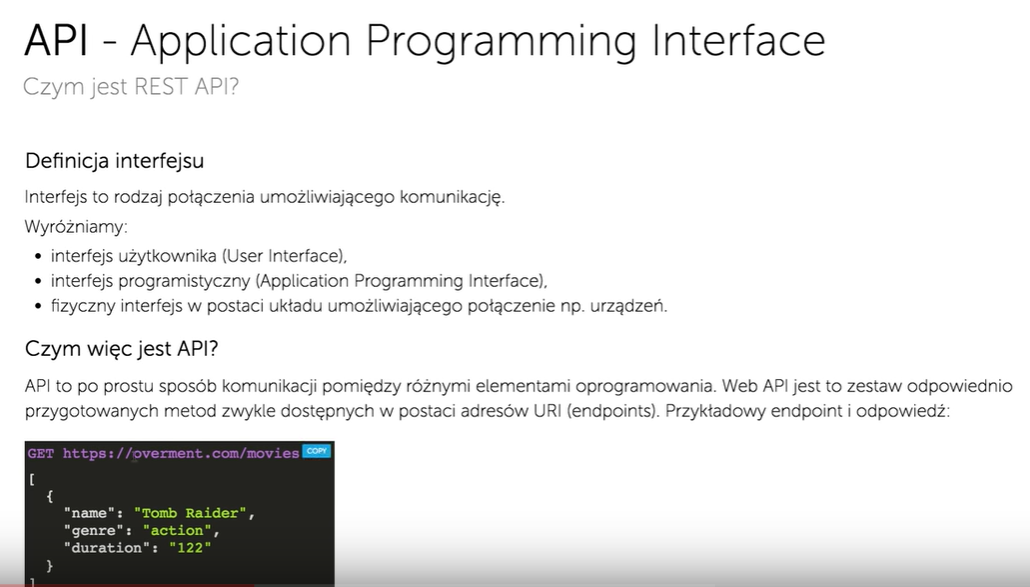
**5. Separacja warstw**

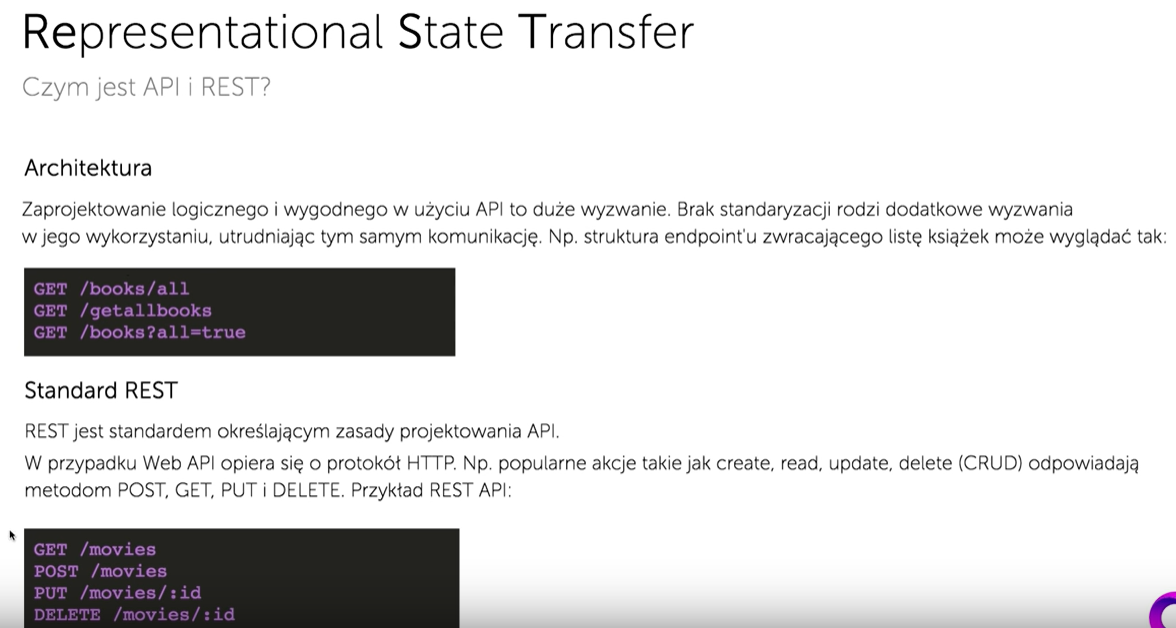
oddzielenie wastw danych, logiki oraz prezentacji

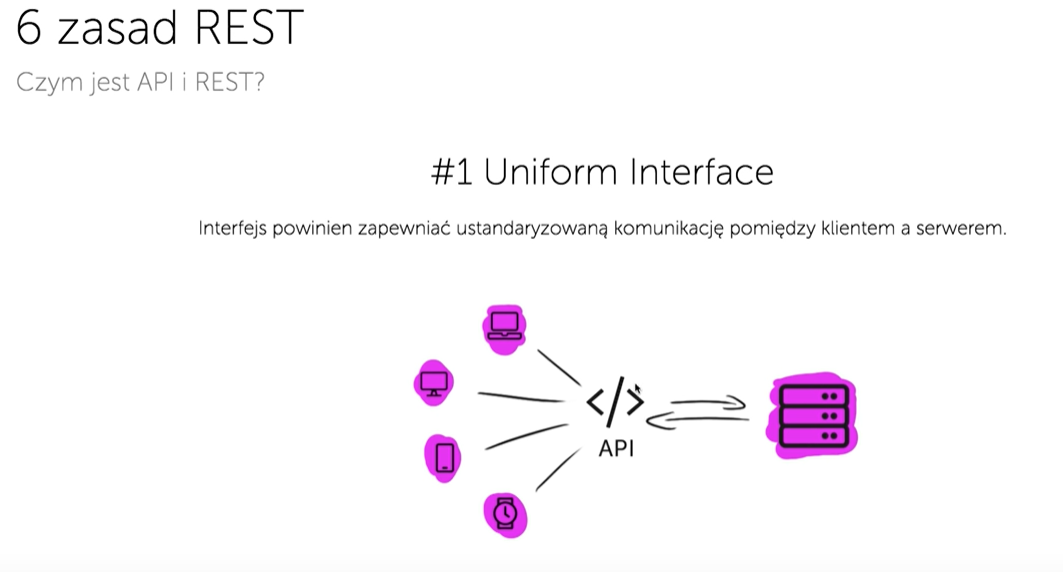
**6. Kod na żądanie (opcjonalnie)**

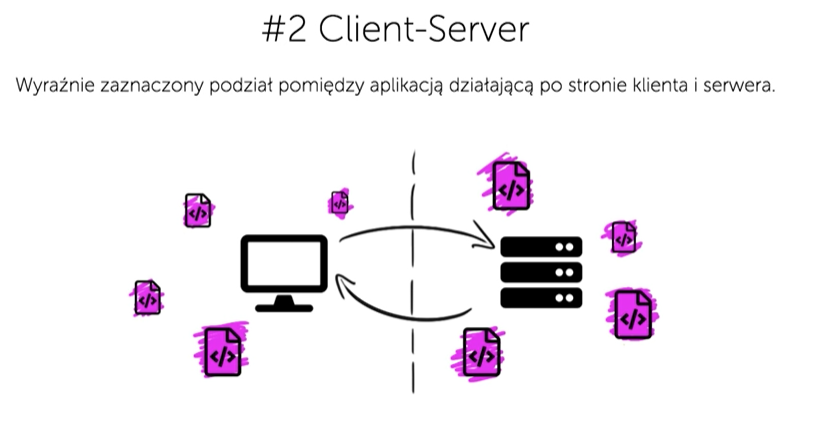
udostępnienie kodu do wykonania po stronie klienta

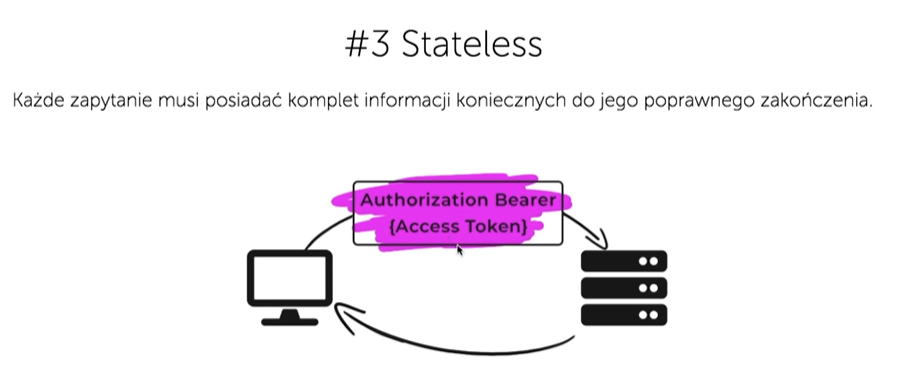
# Przed L3











(serwer nie może przechowywać danych o kliencie)

