

# Comparative analysis of selected programming frameworks dedicated to SPA applications

## Analiza porównawcza wybranych szkieletów dedykowanych budowie aplikacji SPA

Kinga Dybowska\*, Małgorzata Plechawska-Wójcik

*Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland*

### Abstract

The purpose of this study was to compare two popular programming frameworks dedicated to building SPA applications. Two proprietary applications with identical functionalities were analyzed, one created in React and the other in Angular. A ready-made programming tool was used, an original script comparing the speed of adding elements to the database, and the popularity of the technology was checked on the Stack Overflow website and in the Google search engine.

**Keywords:** Angular; React; SPA

### Streszczenie

Celem niniejszej pracy było porównanie dwóch popularnych szkieletów programistycznych dedykowanych budowie aplikacji typu SPA. Analizie zostały poddane dwie autorskie aplikacje o identycznych funkcjonalnościach, jedna z nich powstała w React, natomiast druga w Angularze. Wykorzystano gotowe narzędzie developerskie, autorski skrypt porównujący szybkość dodawania elementów do bazy danych oraz sprawdzono popularność technologii w serwisie Stack Overflow oraz w wyszukiwarce Google.

**Słowa kluczowe:** Angular, React, SPA

\*Corresponding author

Email address: [kinga.mikolajczuk@pollub.edu.pl](mailto:kinga.mikolajczuk@pollub.edu.pl) (K. Dybowska)

©Published under Creative Common License (CC BY-SA v4.0)

## 1. Wstęp

Celem niniejszego artykułu jest analiza różnic i podobieństw wybranych szkieletów programistycznych dedykowanych aplikacjom typu SPA (Single Page Application). SPA w odróżnieniu od innych systemów korzystają tylko z jednego pliku HTML, nie przeładują więc stron podczas ich obsługi. Treści są generowane dynamicznie [1]. Do przeprowadzenia badania stworzone zostały dwie aplikacje, jedna z nich w technologii Angular [2], natomiast druga w React [3]. Powstałe systemy posiadają takie same funkcjonalności oraz podobny wygląd. Analiza została przeprowadzona na podstawie specjalnie przygotowanych scenariuszy badawczych. Do porównania stron wykorzystano gotowe narzędzia developerskie oraz specjalnie napisany skrypt.

Na podstawie celu pracy określono kilka pytań badawczych:

- Który szkielet programistyczny jest szybszy?
- Która aplikacja jest bardziej wydajna?
- Który szkielet programistyczny jest bardziej popularny?

Otrzymane wyniki badań pozwolą odpowiedzieć na pytanie który szkielet programistyczny jest lepszy.

### 1.1. React

React jest biblioteką stworzoną do budowania interfejsów użytkownika. Powstał na podstawie JavaScript,

zaprojektowany przez programistów Meta, dawniej Facebook. Obecnie jest jednym z popularniejszych szkieletów programistycznych wykorzystywanych przy aplikacjach typu SPA. Używa wirtualnego drzewa DOM (Document Object Model) oraz izolowanych komponentów zarządzających własnym stanem. Są one opisane za pomocą JSX (JavaScript XML), jest to połączenie JavaScript oraz HTML (HyperText Markup Language). Dzięki temu możliwe jest wykorzystanie znaczników HTML bezpośrednio w kodzie, zaraz obok logiki aplikacji.

Każdy komponent jest wielokrotnego użytku i posiada cykl życia, czyli montowanie, aktualizację oraz usuwanie. Wymienione etapy posiadają swoje odzwierciedlenie w metodach umożliwiających reakcję na zmiany związane z cyklem życia komponentu [4].

### 1.2. Angular

Biblioteka ta również jest polecana przy tworzeniu stron typu SPA. Jednak w odróżnieniu od React została napisana w języku TypeScript, dzięki czemu łatwiej jest utrzymać czysty i zrozumiały kod. Cechuje się tym, że od razu po uruchomieniu jest gotowa do użycia. Szkielet został opracowany i aktualnie jest wspierany przez Google.

Biblioteka składa się z podstawowych elementów, jakimi są komponenty, zorganizowanych w NgModules. Moduły te zbierają powiązany kod w zestawy funkcjonalne. Aplikacja zawsze posiada co najmniej jeden

moduł, główny, umożliwia on wstępne ładowanie i zazwyczaj posiada kolejne moduły funkcji. Komponenty natomiast definiują widoki, czyli zestawy elementów ekranu. Angular może je wybierać i modyfikować zgodnie z logiką aplikacji. Posiada również mechanizm wiązania danych. Jest to automatyczny sposób aktualizowania widoku przy każdej zmianie modelu. Działa on w dwie strony, odświeża więc też model przy każdej zmianie widoku. Mechanizm ten ułatwia manipulację wirtualnym drzewem DOM [5].

## 2. Przegląd literatury

Analiza dostępnych artykułów o podobnej tematyce pozwoliła wybrać odpowiednie metody badawcze do porównania szkieletów programistycznych.

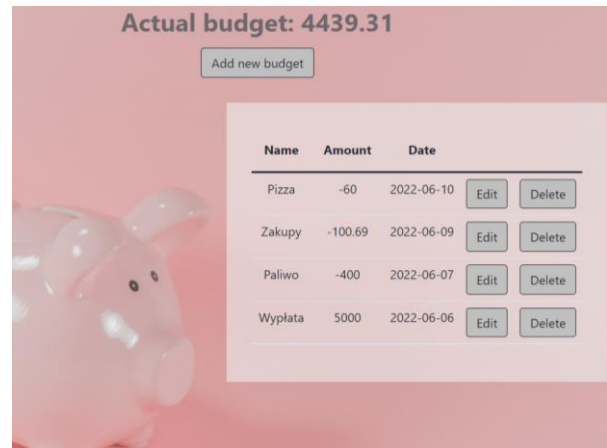
W niektórych pracach do porównania bibliotek wykorzystano średni czas wysyłania żądań do serwera oraz szybkość renderowania komponentów [6-8]. Bardzo wygodnym sposobem przeprowadzenia eksperymentu jest wykorzystanie wirtualnych narzędzi deweloperskich dostępnych online. Dzięki nim można porównać zgodność ze standardami, czas ładowania aplikacji oraz wygodę korzystania na urządzeniach o różnych rozdzielczościach ekranu [9].

Autorzy artykułów o podobnej tematyce często do porównania szkieletów programistycznych wykorzystywali autorskie skrypty [10]. Sprawdzają one między innymi szybkość podstawowych operacji na komponentach, czy też czas ładowania całej aplikacji. Niektóre prace posiadały bardzo dużo scenariuszy badawczych [11]. Oprócz wymienionych wcześniej aspektów w wymienionej pracy sprawdzono również obsługę popularnych zabezpieczeń, obsługę różnych formatów danych, integrację z portalami społecznościowymi takimi jak Facebook, obsługę cache. A także dostęp do baz danych, między innymi Firebase czy SQLite, szybkość współpracy z innymi bibliotekami, takimi jak Google Charts oraz popularność użycia. Efekty eksperymentów zostały przedstawione na wykresach.

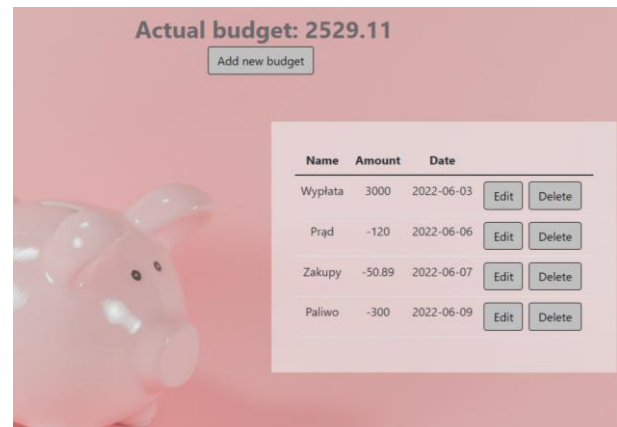
Innym podejściem wykorzystywanym do porównania szkieletów programistycznych było zbadanie szybkości działania aplikacji przy coraz większej liczbie użytkowników jednocześnie korzystających z systemu [12]. Zbadano operacje odczytu danych oraz ich modyfikację. Do tego celu wykorzystano specjalne narzędzie JMeter.

## 3. Metody badawcze

Badanie zostało przeprowadzone na przykładzie prostej aplikacji internetowej umożliwiającej zarządzanie oraz kontrolowanie budżetu domowego. W tym celu powstały dwa systemy, jeden przy użyciu biblioteki React, natomiast drugi przy wykorzystaniu szkieletu programistycznego Angular. Posiadają one identyczne funkcjonalności oraz podobny wygląd. Na Rysunku 1 została zaprezentowana strona stworzona w Angularze, natomiast na Rysunku 2 – w React.



Rysunek 1: Aplikacja Angular.

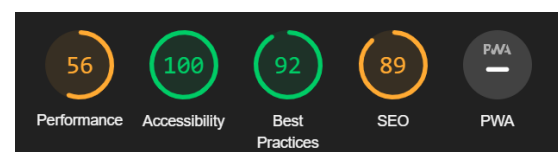


Rysunek 2: Aplikacja React.

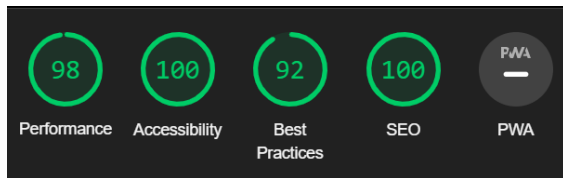
Aby porównać obie strony opracowano specjalne scenariusze badawcze, testując one różne właściwości aplikacji.

### Scenariusz 1: Analiza wydajnościowa z wykorzystaniem narzędzi deweloperskich

Do przeprowadzenia tego badania wykorzystano dedykowane wirtualne narzędzie deweloperskie, dostępne w przeglądarce Google Chrome, jakim jest *Lighthouse* [13]. Pozwala ono przeprowadzić szczegółową analizę strony. Ocenia jej wydajność, dostosowanie do wyszukiwarek oraz potrzeb użytkowników. Pokazuje ono również błędy wykryte podczas pracy aplikacji, a także czas ładowania elementów na stronie. Wyniki analizy przedstawiane są w postaci wykresów. Na Rysunku 3 przedstawiono wyniki analizy dla aplikacji Angular, a na Rysunku 4 dla aplikacji React.



Rysunek 3: Wyniki analizy aplikacji Angular.



Rysunek 4: Wyniki analizy aplikacji React.

Na powyższych rysunkach można zauważyć, że aplikacja React w tym badaniu osiągnęła dużo lepsze wyniki. Wszystkie mierzone statystyki oznaczone zostały kolorem zielonym. Natomiast Angular otrzymał żółte wyniki, oznaczające wartość do poprawy w dwóch przypadkach. Analiza metryk ukazała duże różnice w szybkości działania stron. Angular dużo wolniej łądował aplikację jak i jej pojedyncze elementy, wpłynęło to na wyniki pierwszej metryki. Drugą widoczną różnicą jest SEO, czyli optymalizacja pod kątem wyszukiwarek internetowych. Aplikacja React byłaby wyświetlana wcześniej niż Angular.

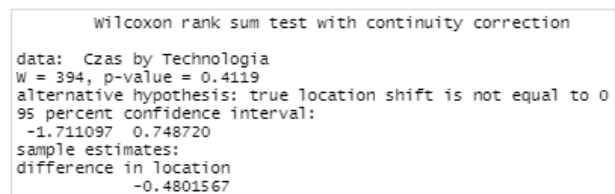
## Scenariusz 2: Analiza wydajnościowa z wykorzystaniem skryptu

Scenariusz ten sprawdza szybkość zapisu określonej liczby rekordów do bazy danych. Do przeprowadzenia tego badania stworzono autorski skrypt, mierzący czas dodawania elementów od momentu naciśnięcia przycisku uruchamiającego formularz, aż do umieszczenia rekordu w bazie. Skrypt został napisany w języku Python, z wykorzystaniem pakietu Selenium. Rozwiązanie to jest bardzo często wykorzystywane do analizy zadań wykonywanych w przeglądarce internetowej.

Pierwsze badanie zostało wykonane dla dodawania kolejno coraz większej liczby rekordów do bazy. Zaczynając od 10, aż do 250, zwiększając liczbę elementów o 10. Podczas ostatniego etapu dodawane jest więc 250 pozycji, a w bazie danych znajduje się już 3000 rekordów. Test ten pozwoli więc zaobserwować działanie aplikacji przy rosnącym obciążeniu bazy danych. Badanie to pokazało różnice w dodawaniu określonej liczby danych. Na początku aplikacje umieszczały rekordy w podobnym czasie. Podczas próby w której testowane było dodawanie 140 elementów Angular zaczął wykonywać zadanie coraz wolniej, natomiast React od tego momentu, aż do dodawania 220 rekordów prezentował dużo mniejszy czas. Jednak w przypadku 230, 240 oraz 250 elementów widać duży spadek jego wydajności, w tych punktach Angular wykonał zadanie znacznie szybciej.

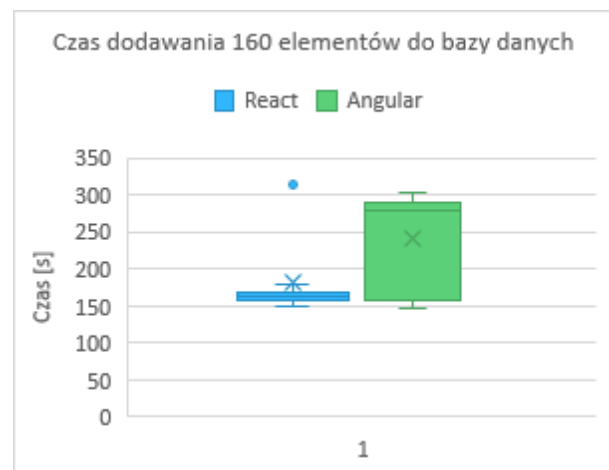
Kolejne badanie jest bardzo podobne do poprzedniego, w tym przypadku jednak do bazy danych dodawane było 20 rekordów, umieszczanie każdej serii danych zostało powtórzone 30 razy. W tym przypadku, po każdej próbie wszystkie elementy były usuwane. Aplikacja za każdym razem działała więc na pustej bazie danych. Dla otrzymanych w ten sposób wyników przeprowadzono test Manna-Whitneya, ponieważ po analizie Shapiro-Wilka okazało się, że dane nie pochodzą z rozkładu normalnego i nie mają rozkładu zbliżonego

do normalnego. Niemożliwe było więc wykonanie badania T Studenta, w związku z tym zdecydowano przeprowadzić test nieparametryczny. Badanie nie wykazało różnic w średnim czasie dodawania elementów do bazy danych. Na rysunku 5 zaprezentowano wyniki testu wykonanego w środowisku *RStudio*. Wartość  $p$  była większa niż, 0.05, co oznacza, że nie było podstaw do odrzucenia hipotezy  $H_0$ , że nie średni czas dodawania rekordów do bazy danych jest porównywalny dla obydwu technologii.



Rysunek 5: Wyniki testu Manna-Whitneya dla 20 elementów.

Ostatnim badaniem wykonanym przy użyciu skryptu było przeprowadzenie identycznego testu jak w poprzednim przypadku, tym razem jednak w bazie danych umieszczane było 160 rekordów. Tutaj również wykorzystano test Manna-Whitneya. Analiza wykazała, że nie ma podstaw do odrzucenia hipotezy  $H_0$ , że średni czas dodawania elementów do bazy danych jest porównywalny dla obydwu aplikacji. Na rysunku 6 zaprezentowano wykres pudełkowy danych wykorzystanych w tym badaniu. Rozmiar pudełka określony jest na podstawie rozstępu ćwiartkowego danych, czyli różnicy między pierwszym, a trzecim kwartylem. Z definicji wynika, że w tym przedziale znajduje się 50% wszystkich obserwacji. Większy rozmiar pudełka świadczy więc o bardziej zróżnicowanych danych. Wąsy, reprezentowane przez linie połączone z pudełkiem oznaczają najmniejszą i największą wartość lub półtorę wartość rozstępu ćwiartkowego. Wartości znajdujące się poza tym zakresem określane są jako punkty odstające. Na wykresie można zauważyć też poziomą linię wewnątrz pudełka, która oznacza medianę. Widoczny jest również znak „x”, który reprezentuje średnią wszystkich danych.

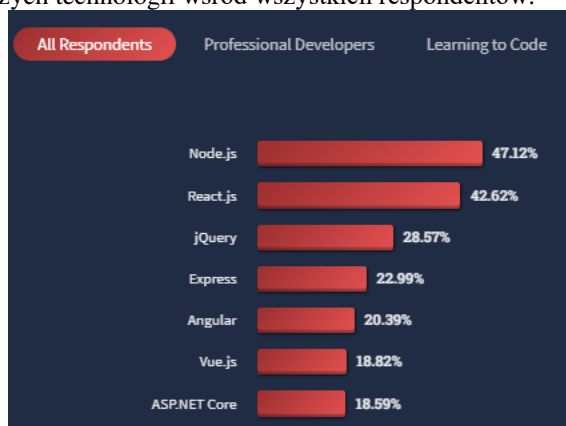


Rysunek 6: Czas dodawania 160 elementów do bazy danych.

Na powyższym rysunku widać duże różnice w średnich czasach dodawania 160 rekordów do bazy danych poszczególnych aplikacji. Zauważalna jest duża różnica w wysokości pudełka reprezentującego wyniki. W przypadku React jest ono kilkukrotnie mniejsze od Angulara. Na tej podstawie oraz analizie średniej i mediany, które również są znacznie mniejsze dla technologii React można stwierdzić, że dodawał on elementy do bazy danych znacznie szybciej niż Angular. Na wykresie widoczne są jednak punkty odstające, które znacznie zawyżają średnią dla aplikacji React. Przez te wartości znalazła się ona poza pudełkiem.

### Scenariusz 3: Popularność użytkowania

W tym przypadku przeanalizowano popularność obydwu technologii, za pomocą jednego z najpopularniejszych serwisów wykorzystywanych przez programistów, jakim jest Stack Overflow [14] oraz narzędzie Google Trends [15]. Ze strony Stack Overflow wykorzystano co roczne badanie pokazujące popularność technologii frontendowych wykorzystywanych przez użytkowników serwisu. Dane pochodzą z 2022 roku. Respondenci zostali podzieleni na grupy, dzięki czemu możliwe jest zobaczenie odpowiedzi dla wszystkich respondentów, profesjonalnych programistów, a także osób uczących się programowania. Na Rysunku 7 przedstawiony został wycinek wykresu, obrazujący siedem najpopularniejszych technologii wśród wszystkich respondentów.



Rysunek 7: Najpopularniejsze technologie według użytkowników Stack Overflow [ <https://survey.stackoverflow.co/2022/#technology-most-popular-technologies> ].

Badanie wykazało, że React był popularniejszy w przypadku wszystkich grup respondentów. Na każdym z wykresów zajmował drugie miejsce z wynikiem powyżej 42%. Natomiast Angular wśród wszystkich respondentów oraz w przypadku profesjonalnych programistów zajmował piątą pozycję z wynikiem ponad 20%. Jednak w przypadku osób uczących się programowania znalazł się dopiero na dziewiątej pozycji. Używało go zaledwie 10% badanych. Wyprzedziły go biblioteki takie jak Django, Flask, Vue.js oraz Next.js. Przeanalizowano również znaczniki, którymi oznaczane są pytania na Stack Overflow. Okazało się, że od 2019

roku hasła związane w *react.js* są znacznie bardziej popularne, niż *angular*.

Drugie narzędzie, jakim jest Google Trends pozwala na porównanie popularności wyszukiwanych haseł przez użytkowników z całego świata. Analizowane dane są frazami wpisywanymi w wyszukiwarce Google w ciągu ostatnich 12 miesięcy na całym świecie. Badanie to wykazało, że hasła związane z React są dużo bardziej popularne. Angular jest popularniejszy jedynie w trzech regionach z 76. We wszystkich pozostałych większym zainteresowaniem cieszy się *React.js*. Największą przewagę ma w Korei Południowej, Chinach, Nigerii, Indonezji oraz Bangladeszu. Proporcje wyszukiwania w tych krajach wynoszą w zaokrągleniu 9:1, na korzyść *React.js*.

### 4. Wnioski

Celem badań, była analiza różnic i podobieństw wybranych szkieletów programistycznych dedykowanych aplikacjom typu SPA. Porównano więc technologie React i Angular na podstawie specjalnie przygotowanych serwisów. Przeprowadzone badania pozwoliły odpowiedzieć na postawione na początku pracy pytania badawcze. Na pytanie: „Która aplikacja jest bardziej wydajna?” pozwoliło odpowiedzieć pierwsze badanie. Wykazało ono, że aplikacja React szybciej ładuje poszczególne elementy oraz ma krótszy czas ładowania strony. Można więc stwierdzić, że jest ona bardziej wydajna. Drugie badanie pozwoliło odpowiedzieć na kolejne pytanie: „Który szkielet programistyczny jest szybszy?”. Analiza wykazała, że nie ma istotnych statystycznie różnic między szybkością dodawania elementów do bazy danych w badanych technologiach. Ostatnie pytanie brzmiało: „Który szkielet programistyczny jest bardziej popularny?”. Ostatnie badanie pokazało, że React cieszy się dużo większą popularnością. W większości regionów na świecie jest dużo częściej wyszukiwany. Programiści chętniej z niego korzystają, jest to zauważalne zwłaszcza wśród osób uczących się programowania. Bardzo rzadko decydują się one na technologię Angular.

Przeprowadzona analiza pozwoliła również odpowiedzieć na główną hipotezę, który szkielet programistyczny jest lepszy. Podsumowując powyższe badania można stwierdzić, że lepszą technologią jest React. Jest bardziej wydajny, szybciej ładuje elementy na stronie oraz cieszy się większą popularnością. Wśród osób uczących się programowania jest zdecydowanie częściej wybierana technologia niż Angular. W przypadku średniego czasu dodawania elementów do bazy danych nie wykryto statystycznie istotnych różnic, jednak wykres pudełkowy dla umieszczania 160 rekordów pokazuje, że średni czas wykonywania tego zadania dla React jest mniej zróżnicowany i zdecydowanie niższy niż w technologii Angular. Wszystkie powyższe badania pozwalają więc stwierdzić, że React jest lepszym szkieletem programistycznym, niż Angular.

**Literatura**

- [1] M. Mikowski, J. Powell, Single Page Web Applications. Programowanie aplikacji internetowych z JavaScript, Helion, 2015.
- [2] A. Freeman, Angular. Profesjonalne techniki programowania. Wydanie II, Helion, 2018.
- [3] P. Kamiński, React. Wstęp do programowania, Helion, 2021.
- [4] E. Porcello, A. Banks, React od podstaw. Nowoczesne wzorce tworzenia aplikacji. Wydanie II, Helion, 2021.
- [5] A. Freeman, Angular. Profesjonalne techniki programowania. Wydanie IV, Helion, 2021.
- [6] R. Baida, M. Andriienko, Analiza porównawcza wybranych technologii internetowych w kontekście tworzenia gier na przykładzie gry planszowej Munchkin, Praca magisterska, Politechnika Lubelska, Lublin 2019.
- [7] K. Boczkowski, Porównanie narzędzi do tworzenia aplikacji typu SPA na przykładzie React i Vue.js, Praca magisterska, Politechnika Lubelska, Lublin 2019.
- [8] M. Jusięga, Analiza porównawcza wybranych wersji szkieletu programistycznego Symphony, Praca magisterska, Politechnika Lubelska, Lublin 2020.
- [9] M. Wrońska, Analiza porównawcza biblioteki jQuery Mobile i frameworka Bootstrap w wytwarzaniu stron responsywnych, Praca magisterska, Politechnika Lubelska, Lublin 2016.
- [10] G. Białycki, B. Pańczyk, Analiza wydajnościowa aplikacji Svelte i Angular, JCSI 19 (2021) 139-143.
- [11] J. Palak, Analiza porównawcza zastosowania szkieletów Angular2 i Ember.js, Praca magisterska, Politechnika Lubelska, Lublin 2017.
- [12] N. Kovalchuk, Porównanie wybranych szkieletów aplikacji internetowych dla technologii Java, Praca magisterska, Politechnika Lubelska, Lublin 2016.
- [13] Opis rozszerzenia Lighthouse, <https://chrome.google.com/webstore/detail/lighthouse/blipmdconlkpinefehnmjammfjpmpbjk?hl=pl>, [27.09.2022].
- [14] Najpopularniejsze technologie według serwisu Stack Overflow, <https://survey.stackoverflow.co/2022/#technology-most-popular-technologies>, [27.09.2022].
- [15] Google Trends popularność haseł Angular i React, <https://trends.google.com/trends/explore?q=%2Fm%2F01211vxv,%2Fg%2F11c6w0ddw9>, [27.09.2022].