

Online Library Testing For Continuous Delivery With Visual Studio 2012 Microsoft Patterns Practices

Testing For Continuous Delivery With Visual Studio 2012 Microsoft Patterns Practices

If you ally infatuation such a referred **testing for continuous delivery with visual studio 2012 microsoft patterns practices** ebook that will provide you worth, acquire the totally best seller from us currently from several preferred authors. If you desire to droll books, lots of novels, tale, jokes, and more fictions collections are afterward launched, from best seller to one of the most current released.

You may not be perplexed to enjoy every books collections testing for continuous delivery with visual studio 2012 microsoft patterns practices that we will certainly offer. It is not approximately the costs. It's not quite what you compulsion currently. This testing for continuous delivery with visual studio 2012 microsoft patterns practices, as one of the most on the go sellers here will utterly be in the course of the best options to review.

GOTO 2016 • Acceptance Testing for Continuous Delivery • Dave Farley *Testing Strategies for Continuous Delivery* Jez Humble | *Continuous Delivery* *Continuous Delivery, Continuous Testing in 3 minutes* Martin Fowler – *Continuous Delivery* *The Foundations of Continuous Delivery* *Continuous Deployment vs Continuous Delivery* Jez Humble: ~~Continuous Delivery – Sounds Great But It Won't Work Here~~ **What is Continuous Delivery?** *What is Continuous Integration?* ~~How to Develop an Agile Testing Strategy for Continuous Delivery~~ *Accelerating*

Online Library Testing For Continuous Delivery With Visual Studio 2012 Microsoft Patterns Practices

Continuous Delivery with Jenkins \u0026amp; ReadyAPI **What is CICD? CI/CD pipelines explained**
What is DevOps? - In Simple English **What's Continuous Integration, Delivery, Deployment? DevOps knowledge [Beyond the Interview]** *Professional Guides: Continuous Integration Continuous Delivery Continuous Integration and Delivery with CircleCI* *What is DevOps? REST API concepts and examples* *Continuous Delivery 101 (Part 1)* *An overview of CI, CD and Jenkins* *Continuous Delivery with Heroku and GitHub* *GTAC 2016: How Flaky Tests in Continuous Integration* *CircleCI Part 1: Introduction to Unit Testing and Continuous Integration* *Continuous Integration, Continuous Deployment (CI-CD) with Azure DevOps* *Dave Farley - Acceptance Testing For Continuous Delivery - PIPELINE Conference 2015* *DevOps: CI/CD Introduction (Continuous Integration, Continuous Delivery, Continuous Deployment)*
Increasing Agility Through Continuous Delivery: Branching Strategy Edition
CI/CD | Continuous Integration | Delivery | Deployment **Testing For Continuous Delivery With**

In a continuous delivery process with efficient testing, the testers work closely with the developers. In fact, the distinction between testers and developers is fast disappearing. The KPIs and targets for testing and quality now belong to the whole development team, rather than just the testing or QA team.

How Does Continuous Delivery Work With Testing? - Testim Blog

Once we have continuous integration and test automation in place, we create a deployment pipeline (the key pattern in continuous delivery). In the deployment pipeline pattern, every change runs a build that a) creates packages that can be deployed to any environment and b) runs unit tests (and possibly other tasks such as static analysis), giving feedback to developers

Online Library Testing For Continuous Delivery With Visual Studio 2012 Microsoft Patterns Practices

in the space of a few minutes.

Continuous Testing - Continuous Delivery

Continuous Testing. Finally comes continuous testing, the process of executing automated tests as part of the software delivery pipeline in order to obtain feedback on the business risks associated with a software release candidate as rapidly as possible. It evolves and extends test automation to address the increased complexity and pace of modern application development and delivery.

Continuous Delivery, Deployment, Integration and Testing ...

Continuous Testing is a software testing type in which the product is evaluated early, often, and throughout the entire Continuous Delivery (CD) process. Continuous testing uses automated tests to ensure teams receive immediate feedback to quickly mitigate as many risks as possible throughout the software development lifecycle.

Continuous Testing Introduction | Benefits, Tools & How to ...

The following are the points that would explain how continuous testing can improve continuous delivery: – Executing API, UI, and Performance testing along with Regression testing will ensure the quality of the app in various customized cases. Continuous testing will allow the developers to perform a live test of the implementation, functionality, and behaviour of their code with testing tools.

Online Library Testing For Continuous Delivery With Visual Studio 2012 Microsoft Patterns Practices

Importance of Continuous Testing in Continuous Delivery ...

Continuous delivery (CD) is all about delivering new code releases as fast as possible to customers. Automated testing is critical to that goal. There's no way to automate delivery to users if there is a manual, time-consuming step within the delivery process. CD is a part of a greater deployment pipeline.

Automated Software Testing for Continuous Delivery

To understand Continuous Testing and Continuous Delivery (CD), it's important to look at the world of DevOps – the culture driving how we build, test, release, and operate software. That culture thrives by focusing on collaboration and communication, flow, feedback, and continuous learning and improvement. So how does testing and CD fit in?

Continuous Testing and Continuous Delivery in a DevOps ...

“Continuous testing is the process of executing automated tests as part of the software delivery pipeline to obtain immediate feedback on the business risks associated with a software release ...

Importance of Continuous Testing In Agile and Continuous ...

Logical concept of continuous delivery is quite similar to the Agile concept of delivery. It is closely related to Continuous Integration & DevOps as well. In other words, continuous delivery can be said as an extended version of agile and continuous integration methodologies. Continuous Delivery is often confused with continuous deployment.

Online Library Testing For Continuous Delivery With Visual Studio 2012 Microsoft Patterns Practices

Top 15 Best Continuous Delivery Tools in 2020 (A Complete ...

If companies are to speed up their app delivery pipeline, they need to remember the success of continuous delivery hinges on effective employment of continuous testing.

A roadmap for continuous delivery and releasing with ...

Continuous delivery is an extension of continuous integration since it automatically deploys all code changes to a testing and/or production environment after the build stage. This means that on top of automated testing, you have an automated release process and you can deploy your application any time by clicking a button.

Continuous integration vs. continuous delivery vs ...

Unlike continuous integration, testing and integrating phases are eliminated and the traditional process of code freeze is followed. Benefits of Continuous Delivery. If the best practices are followed, continuous delivery can help your application development in quite a few ways.

What is Continuous Integration and Continuous Delivery (CI ...

Continuous testing is a critical driver behind the effectiveness of CI/CD (continuous integration/continuous delivery) processes and plays a crucial role in accelerating SDLC timelines by improving code quality, avoiding costly bottlenecks, and expediting DevOps processes.

Online Library Testing For Continuous Delivery With Visual Studio 2012 Microsoft Patterns Practices

What is Continuous Testing? | IBM

Defining a Test Strategy for Continuous Delivery Testing is an important part of building a product right. Continuous Delivery makes that more explicit by building quality in. In this blog post we'll see how you can start off testing on the wrong foot.

Defining a Test Strategy for Continuous Delivery - Simple ...

This diagram shows continuous delivery in a DevOps model with testing everywhere. Lisi makes the key point – success in continuous delivery means shortening feedback loops to learn early. Every point of development and delivery needs validation. In planning – how do you test your plans?

Whole Team Testing for Continuous Delivery - DEV

DevOps is a set of practices that combines software development (Dev) and IT operations (Ops). It aims to shorten the systems development life cycle and provide continuous delivery with high software quality. DevOps is complementary with Agile software development; several DevOps aspects came from Agile methodology.

DevOps - Wikipedia

Continuous delivery is a software engineering approach in which teams produce software in short cycles, ensuring that the software can be reliably released at any time and, when releasing the software, doing so manually. It aims at building, testing, and releasing software with greater speed and frequency. The approach helps reduce the cost, time, and risk of

Online Library Testing For Continuous Delivery With Visual Studio 2012 Microsoft Patterns Practices

delivering changes by allowing for more incremental updates to applications in production. A straightforward and repeatable deployment pro

Continuous delivery - Wikipedia

Integrate your automated tests with your continuous integration pipeline: For end-to-end automation and continuous delivery, it is essential to integrate your testing with the CI/CD pipeline and having a tool that integrates with needed tools readily is the need of the hour. 11.

Winner of the 2011 Jolt Excellence Award! Getting software released to users is often a painful, risky, and time-consuming process. This groundbreaking new book sets out the principles and technical practices that enable rapid, incremental delivery of high quality, valuable new functionality to users. Through automation of the build, deployment, and testing process, and improved collaboration between developers, testers, and operations, delivery teams can get changes released in a matter of hours— sometimes even minutes—no matter what the size of a project or the complexity of its code base. Jez Humble and David Farley begin by presenting the foundations of a rapid, reliable, low-risk delivery process. Next, they introduce the “deployment pipeline,” an automated process for managing all changes, from check-in to release. Finally, they discuss the “ecosystem” needed to support continuous delivery, from infrastructure, data and configuration management to governance. The authors introduce state-of-the-art techniques, including automated infrastructure management and data

Online Library Testing For Continuous Delivery With Visual Studio 2012 Microsoft Patterns Practices

migration, and the use of virtualization. For each, they review key issues, identify best practices, and demonstrate how to mitigate risks. Coverage includes

- Automating all facets of building, integrating, testing, and deploying software
- Implementing deployment pipelines at team and organizational levels
- Improving collaboration between developers, testers, and operations
- Developing features incrementally on large and distributed teams
- Implementing an effective configuration management strategy
- Automating acceptance testing, from analysis to implementation
- Testing capacity and other non-functional requirements
- Implementing continuous deployment and zero-downtime releases
- Managing infrastructure, data, components and dependencies
- Navigating risk management, compliance, and auditing

Whether you're a developer, systems administrator, tester, or manager, this book will help your organization move from idea to release faster than ever—so you can deliver value to your business rapidly and reliably.

Continuous delivery adds enormous value to the business and the entire software delivery lifecycle, but adopting this practice means mastering new skills typically outside of a developer's comfort zone. In this practical book, Daniel Bryant and Abraham Marín-Pérez provide guidance to help experienced Java developers master skills such as architectural design, automated quality assurance, and application packaging and deployment on a variety of platforms. Not only will you learn how to create a comprehensive build pipeline for continually delivering effective software, but you'll also explore how Java application architecture and deployment platforms have affected the way we rapidly and safely deliver new software to production environments. Get advice for beginning or completing your migration to

Online Library Testing For Continuous Delivery With Visual Studio 2012 Microsoft Patterns Practices

continuous delivery Design architecture to enable the continuous delivery of Java applications Build application artifacts including fat JARs, virtual machine images, and operating system container (Docker) images Use continuous integration tooling like Jenkins, PMD, and find-secs to automate code quality checks Create a comprehensive build pipeline and design software to separate the deploy and release processes Explore why functional and system quality attribute testing is vital from development to delivery Learn how to effectively build and test applications locally and observe your system while it runs in production

Janet Gregory and Lisa Crispin pioneered the agile testing discipline with their previous work, *Agile Testing*. Now, in *More Agile Testing*, they reflect on all they've learned since. They address crucial emerging issues, share evolved agile practices, and cover key issues agile testers have asked to learn more about. Packed with new examples from real teams, this insightful guide offers detailed information about adapting agile testing for your environment; learning from experience and continually improving your test processes; scaling agile testing across teams; and overcoming the pitfalls of automated testing. You'll find brand-new coverage of agile testing for the enterprise, distributed teams, mobile/embedded systems, regulated environments, data warehouse/BI systems, and DevOps practices. You'll come away understanding

- How to clarify testing activities within the team
- Ways to collaborate with business experts to identify valuable features and deliver the right capabilities
- How to design automated tests for superior reliability and easier maintenance
- How agile team members can improve and expand their testing skills
- How to plan "just enough," balancing small increments with larger feature sets and the entire system
- How to use testing to identify and

Online Library Testing For Continuous Delivery With Visual Studio 2012 Microsoft Patterns Practices

mitigate risks associated with your current agile processes and to prevent defects • How to address challenges within your product or organizational context • How to perform exploratory testing using “personas” and “tours” • Exploratory testing approaches that engage the whole team, using test charters with session- and thread-based techniques • How to bring new agile testers up to speed quickly—without overwhelming them Janet Gregory is founder of DragonFire Inc., an agile quality process consultancy and training firm. Her passion is helping teams build quality systems. For almost fifteen years, she has worked as a coach and tester, introducing agile practices into companies of all sizes and helping users and testers understand their agile roles. She is a frequent speaker at agile and testing software conferences, and is a major contributor to the agile testing community. Lisa Crispin, an experienced agile testing practitioner and coach, regularly leads conference workshops on agile testing and contributes frequently to agile software publications. She enjoys collaborating as part of an awesome agile team to produce quality software. Since 1982, she has worked in a variety of roles on software teams, in a wide range of industries. She joined her first agile team in 2000 and continually learns from other teams and practitioners.

Getting started with the processes and the tools to continuously deliver high-quality software About This Book Incorporate popular development practices to prevent messy code Automate your build, integration, release, and deployment processes with Jenkins, Git, and Gulp?and learn how continuous integration (CI) can save you time and money Gain an end-to-end overview of Continuous Integration using different languages (JavaScript and C#) and tools (Gulp and Jenkins) Who This Book Is For This book is for developers who want to understand

Online Library Testing For Continuous Delivery With Visual Studio 2012 Microsoft Patterns Practices

and implement Continuous Integration and Delivery in their daily work. A basic knowledge of at least JavaScript and HTML/CSS is required. Knowing C# and SQL will come in handy. Most programmers who have programmed in a (compiled) C-like language will be able to follow along. What You Will Learn Get to know all the aspects of Continuous Integration, Deployment, and Delivery Find out how Git can be used in a CI environment Set up browser tests using Karma and Selenium and unit tests using Jasmine Use Node.js, npm, and Gulp to automate tasks such as linting, testing, and minification Explore different Jenkins jobs to integrate with Node.js and C# projects Perform Continuous Delivery and Deployment using Jenkins Test and deliver a web API In Detail The challenge faced by many teams while implementing Continuous Deployment is that it requires the use of many tools and processes that all work together. Learning and implementing all these tools (correctly) takes a lot of time and effort, leading people to wonder whether it's really worth it. This book sets up a project to show you the different steps, processes, and tools in Continuous Deployment and the actual problems they solve. We start by introducing Continuous Integration (CI), deployment, and delivery as well as providing an overview of the tools used in CI. You'll then create a web app and see how Git can be used in a CI environment. Moving on, you'll explore unit testing using Jasmine and browser testing using Karma and Selenium for your app. You'll also find out how to automate tasks using Gulp and Jenkins. Next, you'll get acquainted with database integration for different platforms, such as MongoDB and PostgreSQL. Finally, you'll set up different Jenkins jobs to integrate with Node.js and C# projects, and Jenkins pipelines to make branching easier. By the end of the book, you'll have implemented Continuous Delivery and deployment from scratch. Style and approach This practical book takes a step-by-step approach to explaining all the

Online Library Testing For Continuous Delivery With Visual Studio 2012 Microsoft Patterns Practices

concepts of Continuous Integration and delivery, and how it can help you deliver a high-quality product.

Get past the myths of testing in agile environments - and implement agile testing the RIGHT way. * * For everyone concerned with agile testing: developers, testers, managers, customers, and other stakeholders. * Covers every key issue: Values, practices, organizational and cultural challenges, collaboration, metrics, infrastructure, documentation, tools, and more. * By two of the world's most experienced agile testing practitioners and consultants. Software testing has always been crucial, but it may be even more crucial in agile environments that rely heavily on repeated iterations of software capable of passing tests. There are, however, many myths associated with testing in agile environments. This book helps agile team members overcome those myths -- and implement testing that truly maximizes software quality and value. Long-time agile testers Lisa Crispin and Janet Gregory offer powerful insights for three large, diverse groups of readers: experienced testers who are new to agile; members of newly-created agile teams who aren't sure how to perform testing or work with testers; and test/QA managers whose development teams are implementing agile. Readers will learn specific agile testing practices and techniques that can mean the difference between success and failure; discover how to transition 'traditional' test teams to agile; and learn how to integrate testers smoothly into agile teams. Drawing on extensive experience, the authors illuminate topics ranging from culture to test planning to automated tools. They cover every form of testing: business-facing tests, technology-facing tests, exploratory tests, context-driven and scenario tests, load, stability, and endurance tests, and more. Using this book's techniques, readers can

Online Library Testing For Continuous Delivery With Visual Studio 2012 Microsoft Patterns Practices

improve the effectiveness and reduce the risks of any agile project or initiative.

For any software developer who has spent days in “integration hell,” cobbling together myriad software components, *Continuous Integration: Improving Software Quality and Reducing Risk* illustrates how to transform integration from a necessary evil into an everyday part of the development process. The key, as the authors show, is to integrate regularly and often using continuous integration (CI) practices and techniques. The authors first examine the concept of CI and its practices from the ground up and then move on to explore other effective processes performed by CI systems, such as database integration, testing, inspection, deployment, and feedback. Through more than forty CI-related practices using application examples in different languages, readers learn that CI leads to more rapid software development, produces deployable software at every step in the development lifecycle, and reduces the time between defect introduction and detection, saving time and lowering costs. With successful implementation of CI, developers reduce risks and repetitive manual processes, and teams receive better project visibility. The book covers How to make integration a “non-event” on your software development projects How to reduce the amount of repetitive processes you perform when building your software Practices and techniques for using CI effectively with your teams Reducing the risks of late defect discovery, low-quality software, lack of visibility, and lack of deployable software Assessments of different CI servers and related tools on the market The book’s companion Web site, www.integratebutton.com, provides updates and code examples.

Online Library Testing For Continuous Delivery With Visual Studio 2012 Microsoft Patterns Practices

Using Continuous Delivery, you can bring software into production more rapidly, with greater reliability. A Practical Guide to Continuous Delivery is a 100% practical guide to building Continuous Delivery pipelines that automate rollouts, improve reproducibility, and dramatically reduce risk. Eberhard Wolff introduces a proven Continuous Delivery technology stack, including Docker, Chef, Vagrant, Jenkins, Graphite, the ELK stack, JBehave, and Gatling. He guides you through applying these technologies throughout build, continuous integration, load testing, acceptance testing, and monitoring. Wolff's start-to-finish example projects offer the basis for your own experimentation, pilot programs, and full-fledged deployments. A Practical Guide to Continuous Delivery is for everyone who wants to introduce Continuous Delivery, with or without DevOps. For managers, it introduces core processes, requirements, benefits, and technical consequences. Developers, administrators, and architects will gain essential skills for implementing and managing pipelines, and for integrating Continuous Delivery smoothly into software architectures and IT organizations. Understand the problems that Continuous Delivery solves, and how it solves them

- Establish an infrastructure for maximum software automation
- Leverage virtualization and Platform as a Service (PAAS) cloud solutions
- Implement build automation and continuous integration with Gradle, Maven, and Jenkins
- Perform static code reviews with SonarQube and repositories to store build artifacts
- Establish automated GUI and textual acceptance testing with behavior-driven design
- Ensure appropriate performance via capacity testing
- Check new features and problems with exploratory testing
- Minimize risk throughout automated production software rollouts
- Gather and analyze metrics and logs with Elasticsearch, Logstash, Kibana (ELK), and Graphite
- Manage the introduction of Continuous Delivery into your enterprise
- Architect software to facilitate Continuous Delivery of

Online Library Testing For Continuous Delivery With Visual Studio 2012 Microsoft Patterns Practices

new capabilities

Learn continuous deployment and automation with code-signing, continuous testing, building, deploying, and releasing of your app. Key Features A practical guide on automating your mobile development pipeline with Fastlane, Jenkins, and Slack. Build, test, run and deploy your mobile application release with this end to end guide. Implement Continuous Integration, delivery, and deployment practices to optimize your application development workflow for faster and efficient release builds. Book Description Competitive mobile apps depend strongly on the development team's ability to deliver successful releases, consistently and often. Although continuous integration took a more mainstream priority among the development industry, companies are starting to realize the importance of continuity beyond integration and testing. This book starts off with a brief introduction to fastlane—a robust command-line tool that enables iOS and Android developers to automate their releasing workflow. The book then explores and guides you through all of its features and utilities; it provides the reader a comprehensive understanding of the tool and how to implement them. Themes include setting up and managing your certificates and provisioning and push notification profiles; automating the creation of apps and managing the app metadata on iTunes Connect and the Apple Developer Portal; and building, distributing and publishing your apps to the App Store. You will also learn how to automate the generation of localized screenshots and mesh your continuous delivery workflow into a continuous integration workflow for a more robust setup. By the end of the book, you will gain substantial knowledge on delivering bug free, developer-independent, and stable application release cycle. What you will learn Harness the fastlane tools for the

Online Library Testing For Continuous Delivery With Visual Studio 2012 Microsoft Patterns Practices

Continuous Deployment strategy Integrate Continuous Deployment with existing Continuous Integration. Automate upload of screenshots across all device screen-sizes Manage push notifications, provisioning profiles, and code-signing certificates Orchestrate automated build and deployments of new versions of your app Regulate your TestFlight users and on-board new testers Who this book is for This book is intended for mobile developers who are keen on incorporating Continuous integration and deployment practices in their workflow.

Unleash the combination of Docker and Jenkins in order to enhance the DevOps workflow About This Book Build reliable and secure applications using Docker containers. Create a complete Continuous Delivery pipeline using Docker, Jenkins, and Ansible. Deliver your applications directly on the Docker Swarm cluster. Create more complex solutions using multi-containers and database migrations. Who This Book Is For This book is indented to provide a full overview of deep learning. From the beginner in deep learning and artificial intelligence to the data scientist who wants to become familiar with Theano and its supporting libraries, or have an extended understanding of deep neural nets. Some basic skills in Python programming and computer science will help, as well as skills in elementary algebra and calculus. What You Will Learn Get to grips with docker fundamentals and how to dockerize an application for the Continuous Delivery process Configure Jenkins and scale it using Docker-based agents Understand the principles and the technical aspects of a successful Continuous Delivery pipeline Create a complete Continuous Delivery process using modern tools: Docker, Jenkins, and Ansible Write acceptance tests using Cucumber and run them in the Docker ecosystem using Jenkins Create multi-container applications using Docker Compose

Online Library Testing For Continuous Delivery With Visual Studio 2012 Microsoft Patterns Practices

Managing database changes inside the Continuous Delivery process and understand effective frameworks such as Cucumber and Flyweight Build clustering applications with Jenkins using Docker Swarm Publish a built Docker image to a Docker Registry and deploy cycles of Jenkins pipelines using community best practices In Detail The combination of Docker and Jenkins improves your Continuous Delivery pipeline using fewer resources. It also helps you scale up your builds, automate tasks and speed up Jenkins performance with the benefits of Docker containerization. This book will explain the advantages of combining Jenkins and Docker to improve the continuous integration and delivery process of app development. It will start with setting up a Docker server and configuring Jenkins on it. It will then provide steps to build applications on Docker files and integrate them with Jenkins using continuous delivery processes such as continuous integration, automated acceptance testing, and configuration management. Moving on you will learn how to ensure quick application deployment with Docker containers along with scaling Jenkins using Docker Swarm. Next, you will get to know how to deploy applications using Docker images and testing them with Jenkins. By the end of the book, you will be enhancing the DevOps workflow by integrating the functionalities of Docker and Jenkins. Style and approach The book is aimed at DevOps Engineers, developers and IT Operations who want to enhance the DevOps culture using Docker and Jenkins.

An airline is supposed to make the experience of booking a flight easy, trouble free, and reliable. But when scheduling software breaks down and flights get canceled, customers will walk, and heads will roll. That's what Leigh Freemark faces the day she and her team launch a software upgrade that fails spectacularly and hits the media immediately. As Senior Director

Online Library Testing For Continuous Delivery With Visual Studio 2012 Microsoft Patterns Practices

of Quality Assurance, her job is to make sure that code is market ready. And she's the one who must face the music when it doesn't. Tasked by senior management to find and fix the source of the failure, Leigh discovers just how essential it has become to radically improve the process of software development by introducing a concept called continuous testing. She must quickly learn what it means, how it works, and how to build it into her company's legacy system. But she soon discovers that managing change is much more difficult than it first appears. The airline business is changing fast, yet old traditions and loyalties still dominate. As she fights to convince her team to change or perish, she discovers that obstructions and opportunities come in surprising forms. *** In The Kitty Hawk Venture, the authors deliver a sound lesson in the importance of continuous testing while taking the reader inside the world of commercial aviation. Each chapter delivers distinct and vital learning opportunities wrapped inside a fast-moving narrative complete with interesting characters, intriguing situations, and even some humor. The book concludes with a "Flight Plan for Continuous Testing" that stands on its own as a valuable resource guide for digital leaders in their continuous testing journey. The story is immediately relatable to anyone who has worked in software development or for the companies that rely on it. Who This Book Is For C-level executives, VPs of apps and quality, VPs of DevOps, architecture and strategy managers, and SMB and enterprise professionals

Copyright code : aa386a50beadc6b6d666bfcba1abe7ee