

# How Developers use API docs

Anthony T. Sansone | COM545

# Based on research paper

## Merseburg University of Applied Sciences

Michael Meng



Stephanie Steinhardt



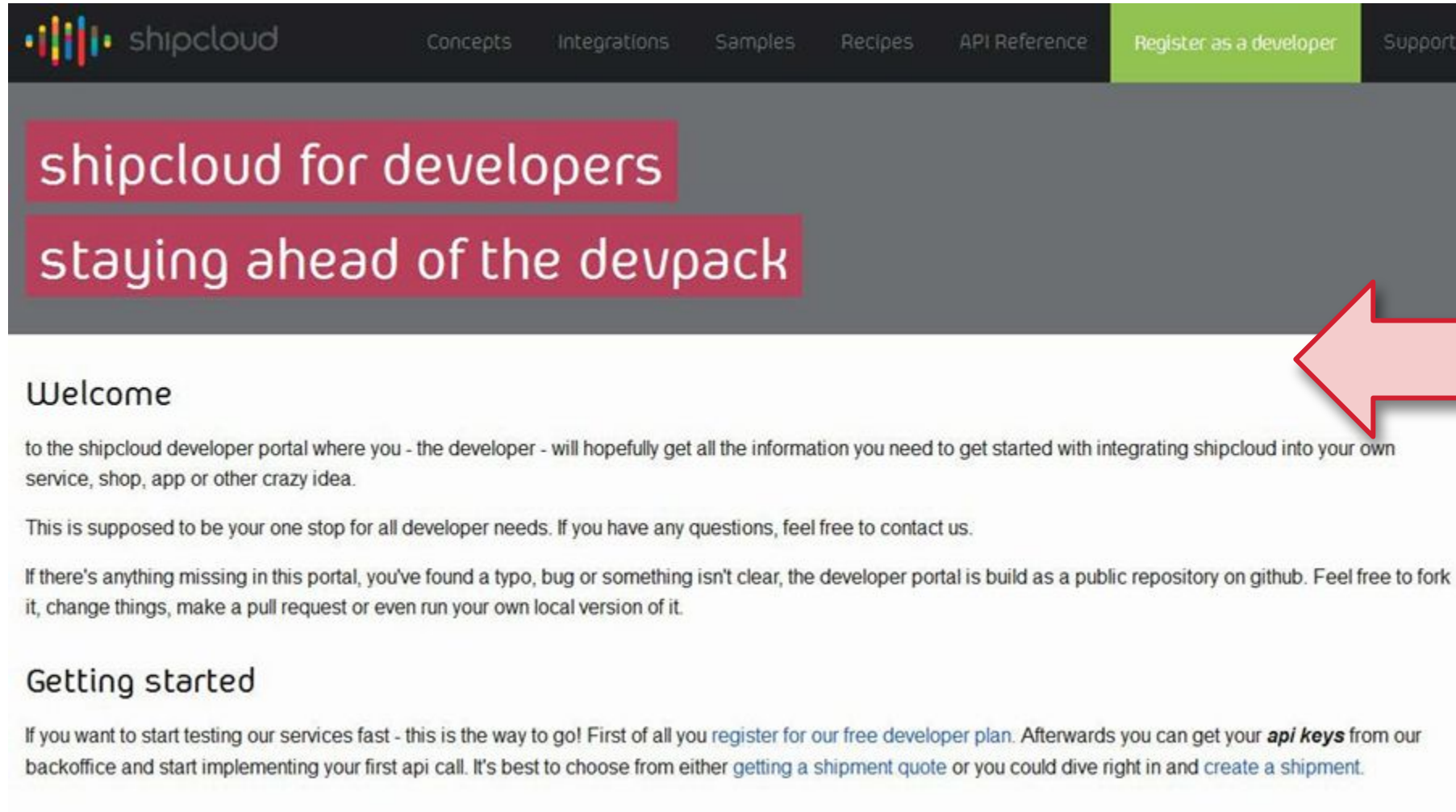
Andreas Schubert



# API use increases, but learning paths challenge developers



# Issues with learning resources may be usability issues



The screenshot shows the shipcloud developer portal. The top navigation bar includes links for Concepts, Integrations, Samples, Recipes, API Reference, Register as a developer (highlighted in green), and Support. The main header area contains the text "shipcloud for developers" and "staying ahead of the devpack" in white text on a dark background. Below this, the "Welcome" section states: "to the shipcloud developer portal where you - the developer - will hopefully get all the information you need to get started with integrating shipcloud into your own service, shop, app or other crazy idea." It also mentions: "This is supposed to be your one stop for all developer needs. If you have any questions, feel free to contact us." and "If there's anything missing in this portal, you've found a typo, bug or something isn't clear, the developer portal is build as a public repository on github. Feel free to fork it, change things, make a pull request or even run your own local version of it." The "Getting started" section says: "If you want to start testing our services fast - this is the way to go! First of all you [register for our free developer plan](#). Afterwards you can get your **api keys** from our backoffice and start implementing your first api call. It's best to choose from either [getting a shipment quote](#) or you could dive right in and [create a shipment](#)." A red box with the text "Show me the API!" and a red arrow points to the "API Reference" link in the navigation bar.

shipcloud

Concepts Integrations Samples Recipes API Reference Register as a developer Support

shipcloud for developers

staying ahead of the devpack

## Welcome

to the shipcloud developer portal where you - the developer - will hopefully get all the information you need to get started with integrating shipcloud into your own service, shop, app or other crazy idea.

This is supposed to be your one stop for all developer needs. If you have any questions, feel free to contact us.

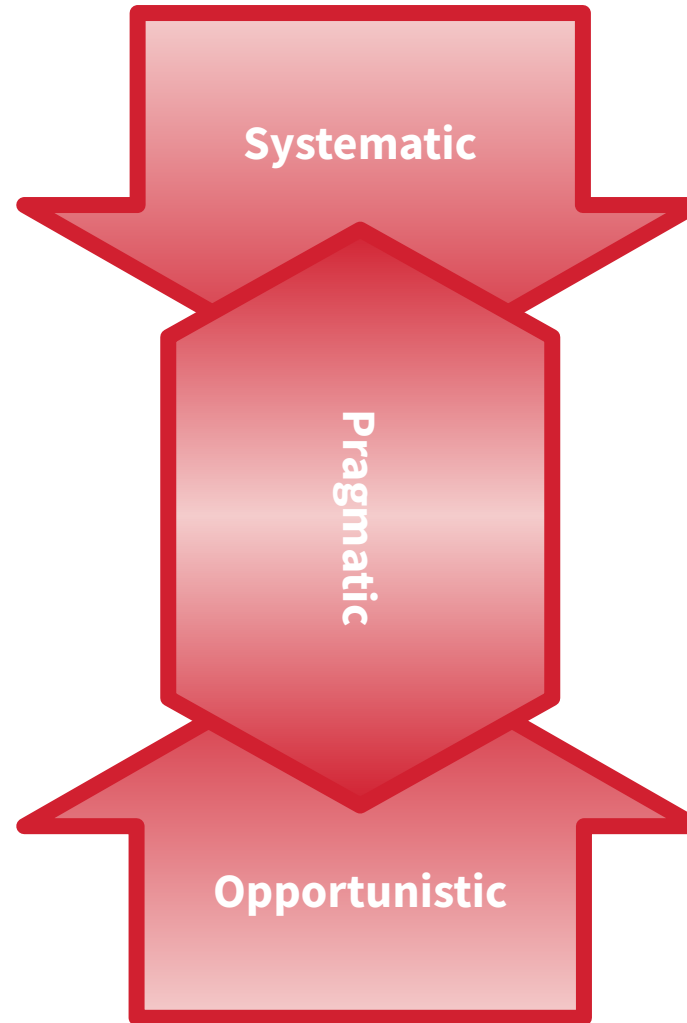
If there's anything missing in this portal, you've found a typo, bug or something isn't clear, the developer portal is build as a public repository on github. Feel free to fork it, change things, make a pull request or even run your own local version of it.

## Getting started

If you want to start testing our services fast - this is the way to go! First of all you [register for our free developer plan](#). Afterwards you can get your **api keys** from our backoffice and start implementing your first api call. It's best to choose from either [getting a shipment quote](#) or you could dive right in and [create a shipment](#).

Show me  
the API!

# Developers use 1 of 3 strategies to learn a new API





# Developers expect accuracy, clarity, completeness in docs



# Devs need concepts, specs, examples, purpose, use cases

The screenshot displays the shipcloud API documentation website. The header includes the shipcloud logo, a search bar, and links for Developer Support, Help Center, and Login. A left sidebar contains a navigation menu with categories like Concepts, Integrations, Carriers, Examples, API Reference, and OpenAPI Spec. Under the API Reference section, a list of topics is shown, including API Introduction, Authentication, Rate Limiting, Parameters, Sanitization, Versioning, Address handling, Label format, Multi package shipments, Metadata, Timeouts, Sandbox vs. Production, Keys, Shipment prices, Cleanup, and Plan up-/downgrades. The main content area is titled 'API Introduction' and provides an overview of the shipping service provider. It explains that shipcloud is a shipping service provider that makes it easy for developers to integrate shipping using one of the major carriers on the German market into their own software, onlineshop or ERP solution. It mentions that they've basically built a wrapper around the carriers' webservices so you won't have to integrate each and every carrier by yourself. It also states that to make it easier for developers, they've created the shipcloud API deliberately using the RESTful architectural style. This means if you're familiar with REST and using RESTful services, you will have no problems using our API. As it's common when implementing a REST-API we're using resource-oriented URLs and HTTP authentication. The 'Authentication' section follows, explaining that for developers to test their code before shipping it, they're supplying every account with test and live API keys. Every call to their API is secured by an API key. This key is tied specifically to their account and therefore should never be given to anyone else outside of their control. You are able to manage your API keys from your account page. It then states that to use the API key you'll have to send it with every request being made. Authentication to their API has to be done via HTTP Basic Auth. You'll have to provide your API key as the basic auth username. You don't have to provide a password. A note advises that the API key has to be Base64 encoded. Finally, it mentions that aside from sending your API key all API requests must be made using HTTPS. Don't bother trying to use HTTP. It will fail! The footer includes status indicators (All Systems Operational), contact information (developers@shipcloud.io), social media links (shipcloud, shipcloud\_devs), a Privacy Statement link, and a Register as a developer button.

shipcloud

Search

Developer Support Help Center Login

Concepts

Integrations

Carriers

Examples

API Reference

OpenAPI Spec

API

Introduction

Authentication

Rate Limiting

Parameters

Sanitization

Versioning

Address handling

Label format

Multi package shipments

Metadata

Timeouts

Sandbox vs. Production

Keys

Shipment prices

Cleanup

Plan up-/downgrades

## API

### Introduction

shipcloud is a shipping service provider that makes it easy for developers to integrate shipping using one of the major carriers on the German market into their own software, onlineshop or ERP solution. We've basically built a wrapper around the carriers' webservices we support so you won't have to integrate each and every carrier by yourself.

To make it easier for you the developer, we've created the shipcloud API deliberately using the RESTful architectural style. This means if you're familiar with REST and using RESTful services, you will have no problems using our API. As it's common when implementing a REST-API we're using resource-oriented URLs and HTTP authentication.

### Authentication

For developers to test their code before shipping it, we're supplying every account with test and live API keys. Every call to our API is secured by an API key. This key is tied specifically to your account and therefore should never be given to anyone else outside of your control. You are able to manage your API keys from your account page.

To use the API key you'll have to send it with every request being made. Authentication to our API has to be done via **HTTP Basic Auth**. You'll have to provide your API key as the basic auth username. You don't have to provide a password.

**Be advised:** The API key has to be **Base64** encoded.

Aside from sending your API key all API requests must be made using HTTPS. Don't bother trying to use HTTP. It will fail!

All Systems Operational developers@shipcloud.io shipcloud shipcloud\_devs Privacy Statement Register as a developer

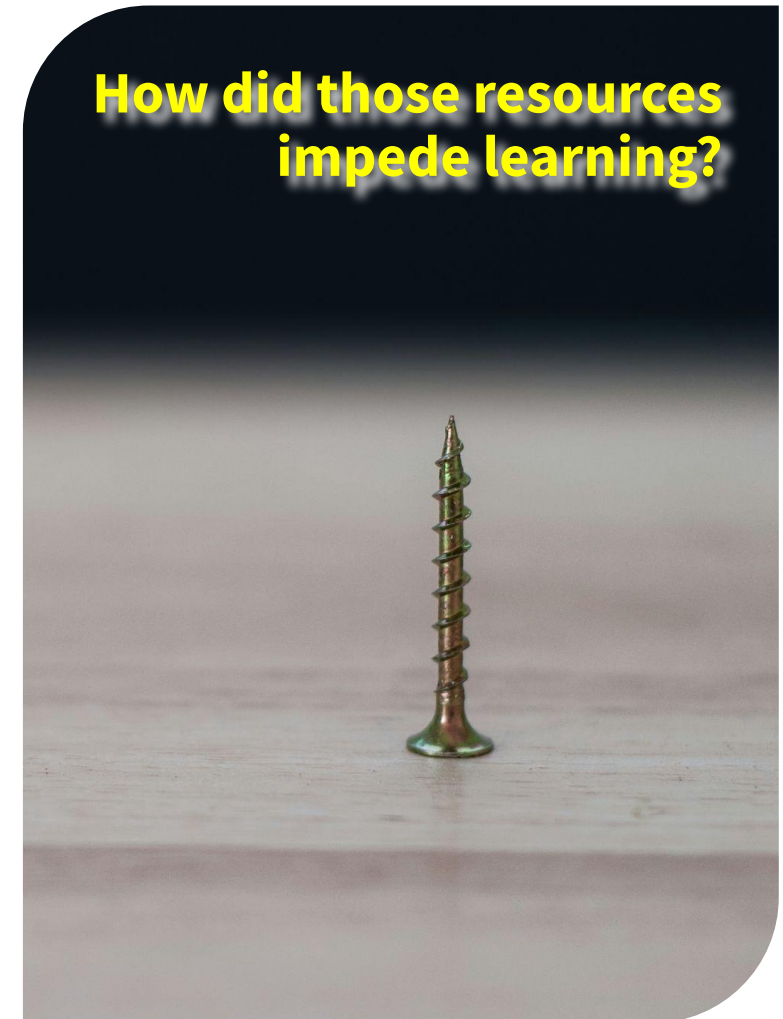


# Studies relied on self reporting, not observations





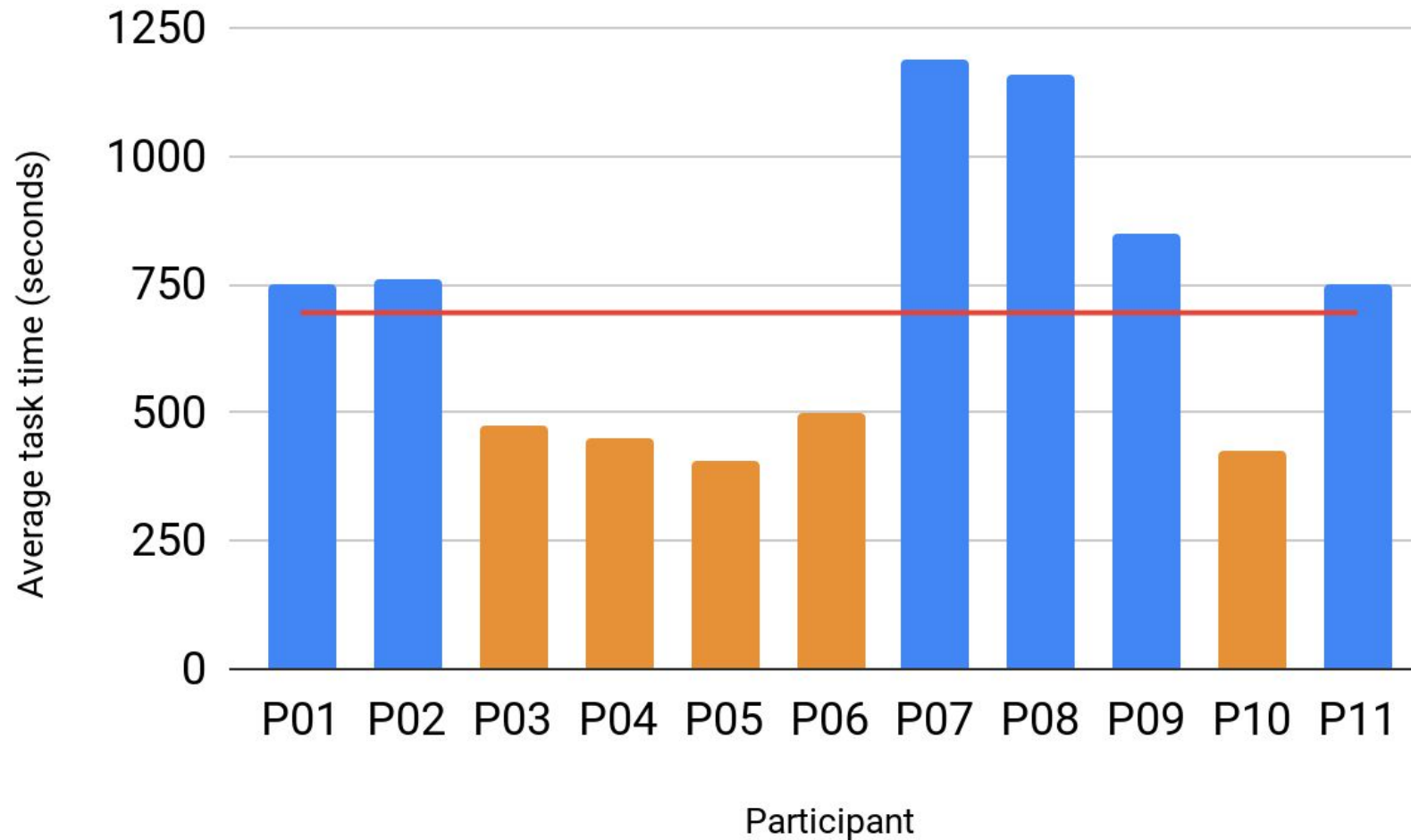
# Observe how devs learn a new API



# Observed 11 devs complete tasks with new API

1. Created API-related tasks
2. Recruited participants and received consent
3. Gathered demographic data
4. Ran and recorded test sessions with tracked eye movement
5. Gave questionnaire for participant feedback
6. Analyzed test sessions and aggregated feedback

# Grouped participants by average task completion time

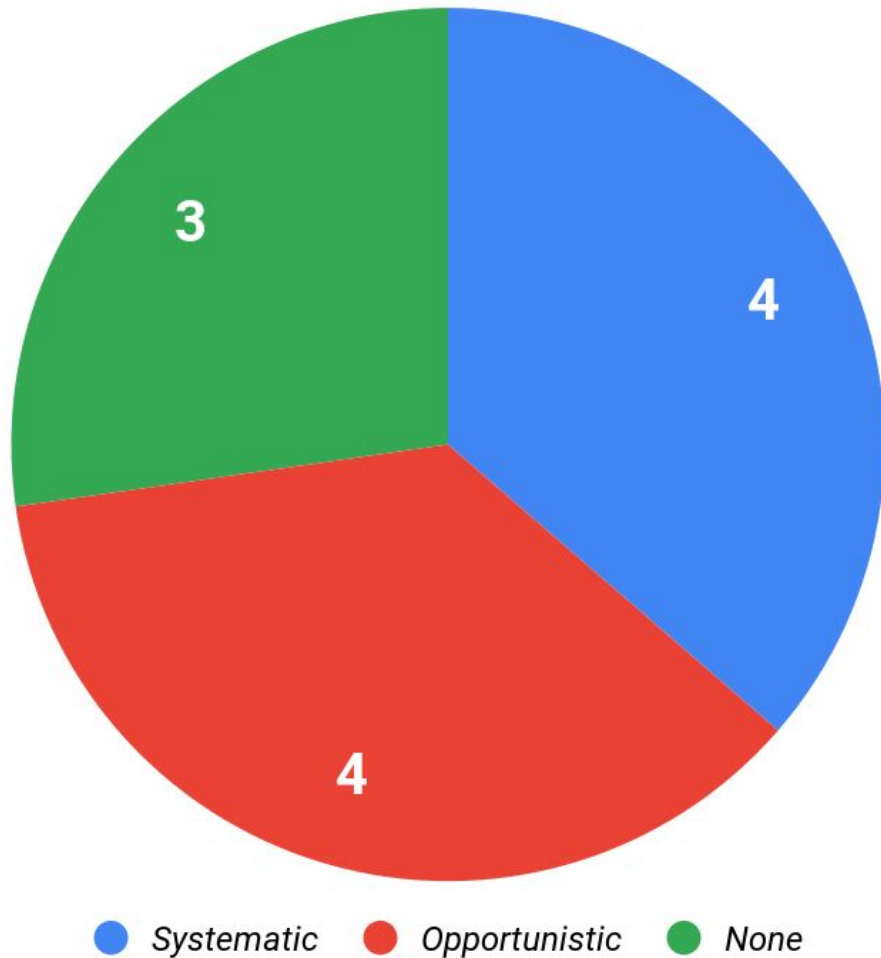


Made 2 groups:

- **Slow:** Time > Mean
- **Fast:** Time < Mean



# Grouped participants (again) by task strategy



## Opportunistic developers characteristics

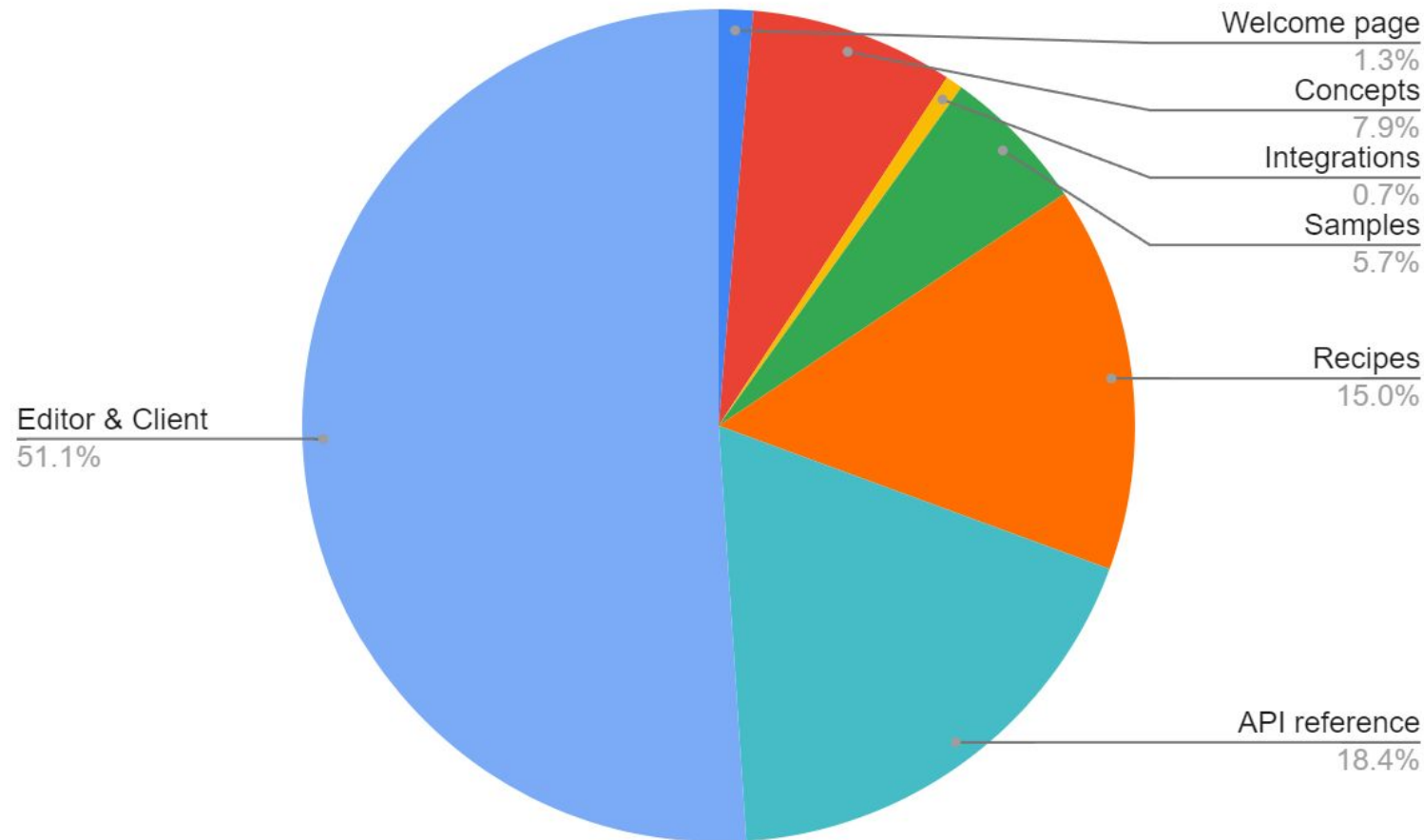
Participant	Speed	Avg Per Task Time	Tasks Completed	Dev Exp	REST Exp Rating	Field	Role
P02	Slow	760 s	5	10 yrs	5	General IT Services	Developer
P03*	Fast	475 s	3	20 yrs	4	E-Commerce	Team lead
P09	Slow	850 s	5	25 yrs	2	Publishing	DB specialist
P10	Fast	425 s	5	11 yrs	3	E-Commerce	Team lead

## Systematic developers characteristics

Participant	Speed	Avg Per Task Time	Tasks Completed	Dev Exp	REST Exp Rating	Field	Role
P4	Fast	450	5	5 yrs	2	E-Commerce	Developer
P5	Fast	405	5	6 yrs	5	E-Commerce	Developer
P7	Slow	1190	3	8 yrs	5	Publishing	Developer
P8	Slow	1160	3	10 yrs	3	Publishing	Developer

# Developers used docs less than half of their time

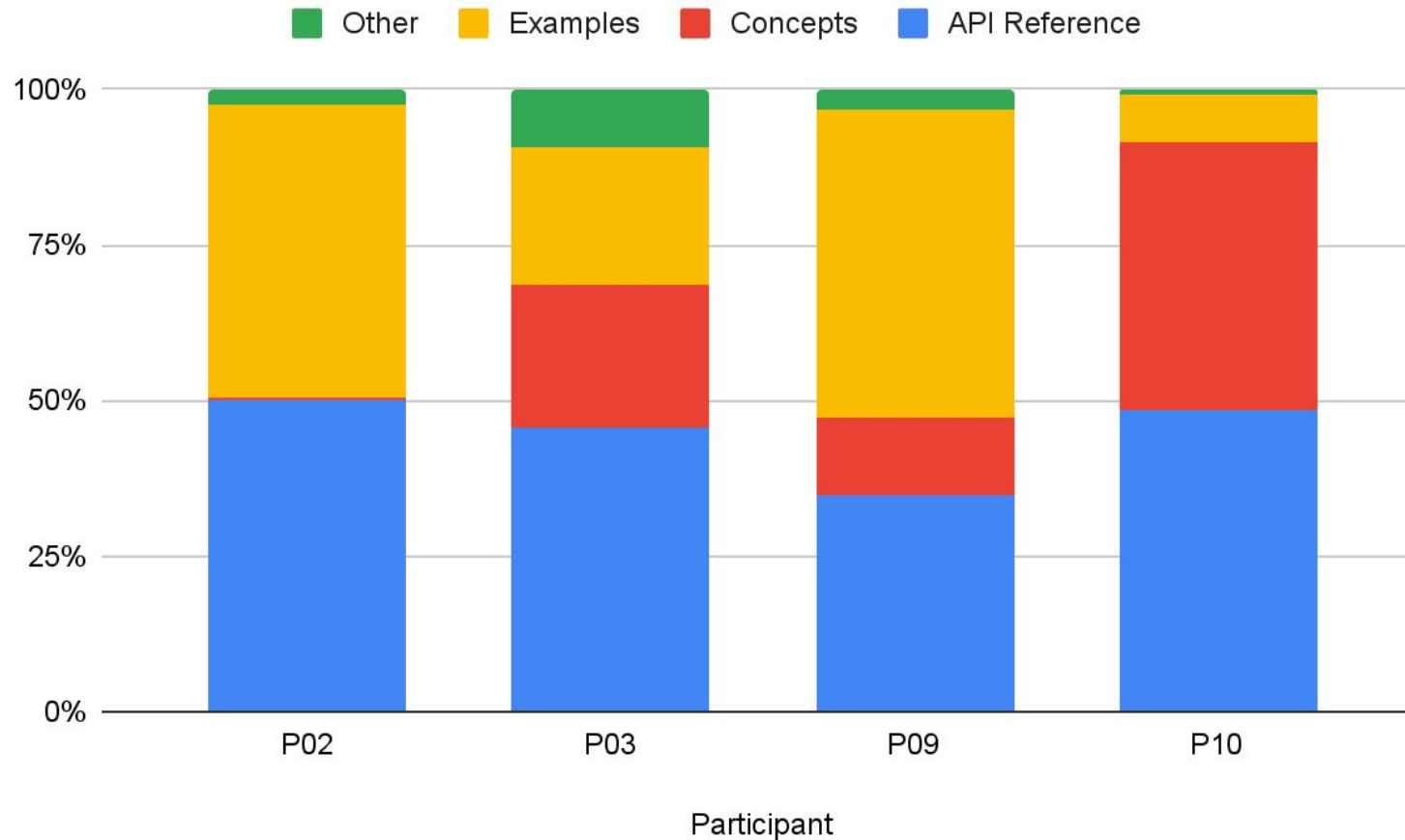
Spent most time in tools, then in reference docs, then in code samples



- Time spent outside of docs > 50%
- Time spent in docs mostly in code examples and spec

# Opportunistic developers explored tasks, made errors

Seasoned developers experimented, skimmed documentation



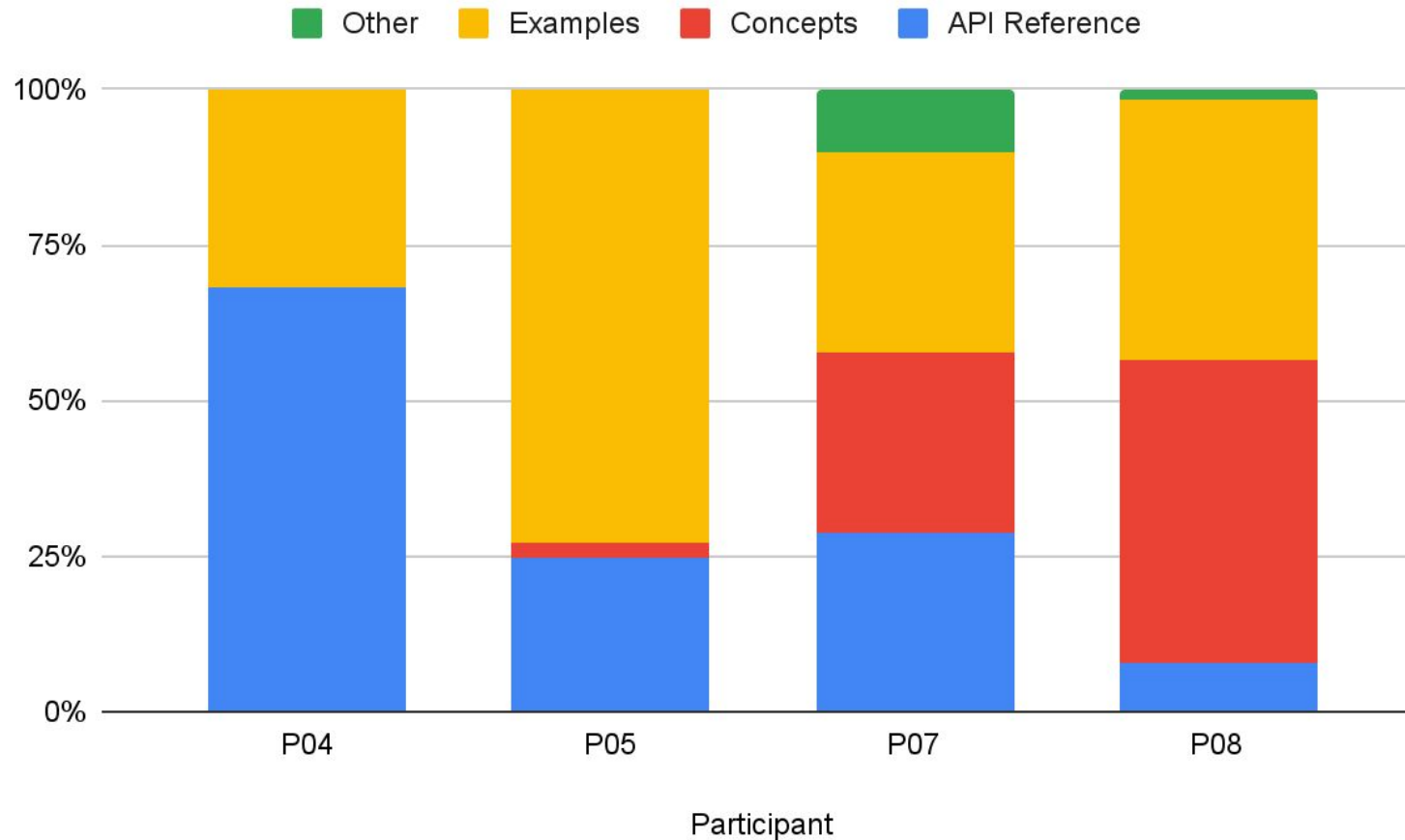
All participants used examples and reference

Some participants used concepts



# Systematic developers explored API, prepared environment

Mid-level developers planned task first, read docs with care



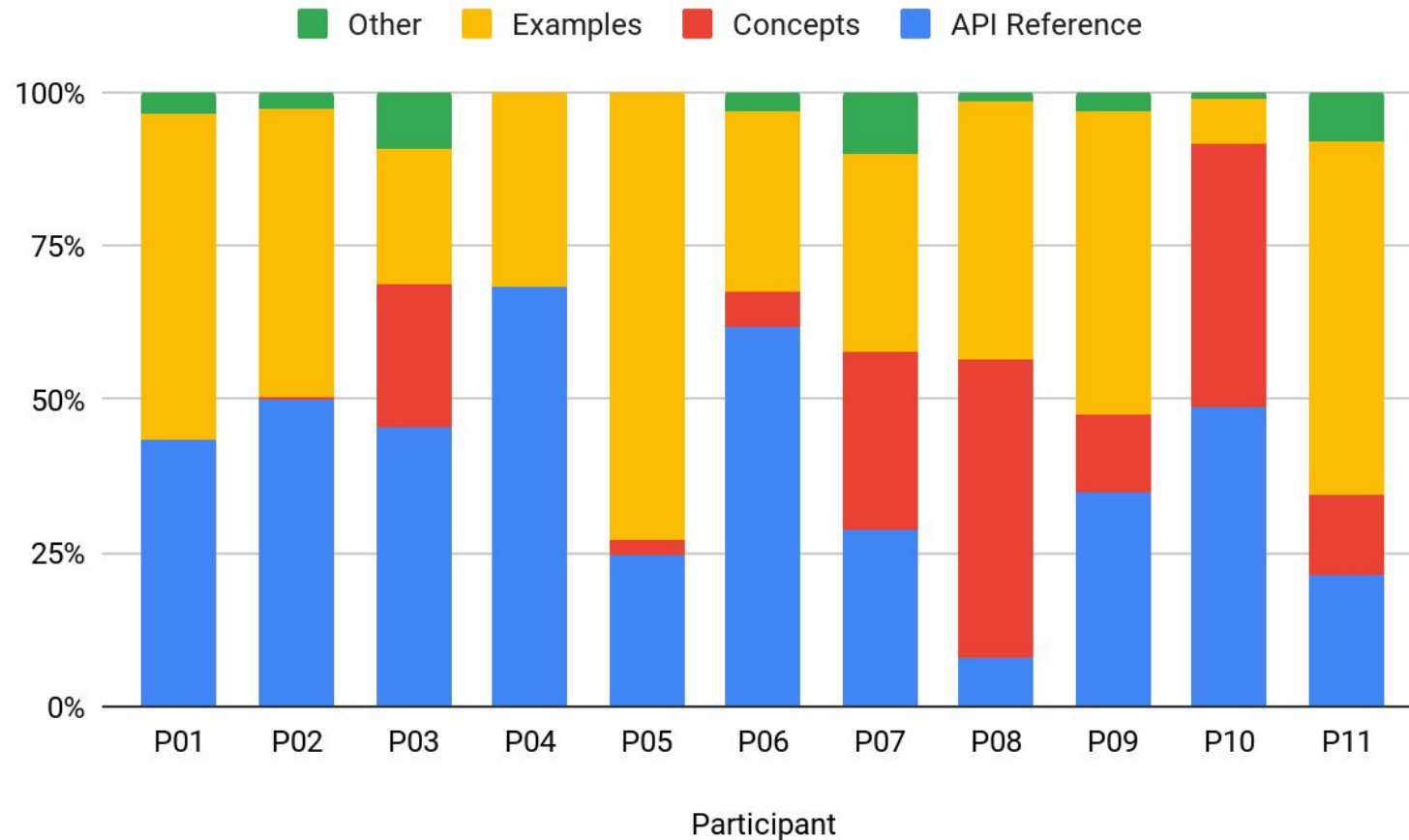
All participants used examples more than 30% of the time

All participants used reference with time spent in a range of 60 pts

Most used concepts with time spent in a range of 49 pts

# Developers varied widely in docs usage

No one part was used exclusively more than another across all developers



All participants used examples and reference

Most participants used concepts

Time spent in topic ranged from 8% to 73% across all participants

# Developers perceived barriers to their tasks

Inconsistent nav, bad structure, no search, and code reuse impeded progress

The screenshot shows the shipcloud developer portal. The navigation bar includes links for Concepts, Integrations, Samples, Recipes, API Reference, Register as a developer, and Support. A red box labeled 'Nav?' points to the navigation links. A red box labeled 'Search?' points to the 'Support' link. A red box labeled 'Starting Point?' points to the 'Welcome' section. The main content area has a dark header with the text 'shipcloud for developers' and 'staying ahead of the devpack'. Below this, the 'Welcome' section contains the following text:

Welcome

to the shipcloud developer portal where you - the developer - will hopefully get all the information you need to get started with integrating shipcloud into your own service, shop, app or other crazy idea.

This is supposed to be your one stop for all developer needs. If you have any questions, feel free to contact us.

If there's anything missing in this portal, you've found a typo, bug or something isn't clear, the developer portal is build as a public repository on github. Feel free to fork it, change things, make a pull request or even run your own local version of it.

**Getting started**

If you want to start testing our services fast - this is the way to go! First of all you [register for our free developer plan](#). Afterwards you can get your **api keys** from our backoffice and start implementing your first api call. It's best to choose from either [getting a shipment quote](#) or you could dive right in and [create a shipment](#).



# API docs need to speed access to relevant content

Need docs to make connections between text and code while making both distinct

The screenshot shows the Shipcloud API documentation interface. The left sidebar contains a navigation menu with the following items: Concepts, Integrations, Carriers (highlighted in red), Examples, API Reference, and OpenAPI Spec. Below this, there are sections for 'Ship Webservice' (listing Services: Standard, One day, One day early, Returns, Expedited, Express 12:00; Package types: Parcel; Additional services: Carbon neutral, Cash on delivery, Adult signature, Saturday delivery, Access point notificati..., Direct delivery only, Signature required), 'Ship Webservice', 'Services', 'Standard', 'One day', 'One day early', 'Returns', 'Expedited', 'Express 12:00', 'Package types', 'Parcel', 'Additional services', 'Carbon neutral', 'Cash on delivery', 'Adult signature', 'Saturday delivery', 'Access point notificati...', 'Direct delivery only', and 'Signature required'. The main content area displays the 'Ship Webservice' details, including a 'Show example' button and a code block for a POST request to `https://api.shipcloud.io/v1/shipments`. The code block contains a JSON payload with 'from' and 'to' information. The footer includes status indicators (All Systems Operational), contact information (developers@shipcloud.io), social media links (shipcloud, shipcloud\_devs), a Privacy Statement link, and a Register as a developer button.

**Annotations:**

- Clarify navigation:** Points to the left sidebar navigation menu.
- Organize by function:** Points to the 'Ship Webservice' section in the sidebar.
- Add search:** Points to the search bar at the top of the main content area.
- Connect concepts to task & code:** Points to the 'Show example' button and the code block.
- Make code examples distinct:** Points to the code block.

# API docs need to relate to real-world scenarios

Need to connect API to real-world activities

The screenshot shows the Shipcloud website's API documentation for DHL v3.1. The left sidebar contains a navigation menu with categories like Concepts, Integrations, Carriers, Examples, API Reference, and OpenAPI Spec. Under 'API Reference', 'DHL Api v3.1' is selected, showing sub-items: Carrier specifics, General information, Services, Standard, Returns, Europaket, Prio, Warenpost, Package types, Bulk shipments, Parcel, Additional services, Advance notice, Delivery, and Print. The main content area displays 'DHL Api v3.1' and 'Carrier specifics' under 'General information'. It explains that the DHL webservice now allows specifying an address by only transmitting the 'street' attribute, which contains both streetname and housenumber. A 'Notice' states that DHL has a separate returns webservice. Below this, 'Additional delivery details in tracking events' are described, with an example JSON response showing a delivered shipment with details like location, status, and delivery details. A red box on the right side of the page, labeled 'Add background knowledge', has an arrow pointing to the text about the 'street' attribute. Another red box on the left, labeled 'Offer quick API access', has an arrow pointing to the 'API Reference' section in the sidebar. A third red box at the bottom left, labeled 'Offer clear examples', has an arrow pointing to the JSON example.

Offer quick API access

Offer clear examples

Add background knowledge

# Fix usability and improve learning and adoption

Docs issues appear to be UX issues

The screenshot shows the Shipcloud developer portal. At the top is a dark navigation bar with the Shipcloud logo and links for Concepts, Integrations, Samples, Recipes, API Reference, Register as a developer (highlighted in green), and Support. Below the navigation bar is a large grey banner with the text "shipcloud for developers" and "staying ahead of the devpack" in white. The main content area has a "Welcome" section with a paragraph about the developer portal and a "Getting started" section with instructions on how to register and get API keys. On the left side of the "Getting started" section is a sidebar with a list of links: Production, Keys, Shipment prices, Cleanup, and Plan up-/downgrades. The main content area also contains text about authentication via HTTP Basic Auth and a note about Base64 encoding API keys. At the bottom of the page is a footer with status indicators, contact information, and a "Register as a developer" button.

shipcloud

Concepts Integrations Samples Recipes API Reference Register as a developer Support

shipcloud for developers

staying ahead of the devpack

## Welcome

to the shipcloud developer portal where you - the developer - will hopefully get all the information you need to get started with integrating shipcloud into your own service, shop, app or other crazy idea.

This is supposed to be your one stop for all developer needs. If you have any questions, feel free to contact us.

If there's anything missing in this portal, you've found a typo, bug or something isn't clear, the developer portal is build as a public repository on github. Feel free to fork it, change things, make a pull request or even run your own local version of it.

## Getting started

If you want to start testing our services fast - this is the way to go! First of all you [register for our free developer plan](#). Afterwards you can get your **api keys** from our backoffice and start implementing your first api call. It's best to choose from either [getting a shipment quote](#) or you could dive right in and [create a shipment](#).

Production  
Keys  
Shipment prices  
Cleanup  
Plan up-/downgrades

made. Authentication to our API has to be done via **HTTP Basic Auth**. You'll have to provide your API key as the basic auth username. You don't have to provide a password.

Be advised: The API key has to be **Base64** encoded.

Aside from sending your API key all API requests must be made using **HTTPS**. Don't bother trying to use **HTTP**. It will fail.

Register as a developer