

# Index

[General Criteria](#)

[Evaluation Criteria](#)

[Rewards Criteria](#)

[Guidelines for Technical Documentation](#)

[Guidelines for Written Content](#)

[Guidelines for Video Content](#)

## General Criteria

- Only **written guides or tutorials** will be accepted. These must showcase a working dApp, technical concept, or feature that can be implemented by Rootstock developers.
- Each submission must include a **GitHub repository** containing a README section with a clear explanation and setup instructions.
- Submissions must be original, technically accurate, and not previously published elsewhere.
- Once approved, you will be invited as a collaborator to publish your educational content on the [Rootstock Community blog](#). Contributions become eligible for rewards only after successful publication.
- If you are contributing technical documentation to the [Rootstock DevPortal](#), it must be submitted directly as a Pull Request (PR) on the repository. Include the PR link in your submission form, ensure the “hactivator” label is applied, and follow the provided template. Submissions consisting of only a single line of code or minor edits that do not offer real functionality or improvement will not be accepted.

## Evaluation Criteria

### 1. Content Depth and Detail

- **Word Count / Length Appropriateness:**  
Content should meet a minimum word count requirement based on the type of contribution:
  - **Short Posts (500 - 800 words):** Suitable for introductory content or narrow topics.
  - **Standard Posts (800 - 1,500 words):** For more detailed guides and tutorials.
  - **In-Depth Tutorials / Case Studies (1,500+ words):** For complex, multi-step guides or advanced topics.
- **Video Length Appropriateness**

Similarly, videos should align with the depth of the content and the audience's expectations:

- **Short Videos (1 - 3 minutes):** Perfect for quick introductions, overviews, or teasers for a larger topic.
- **Standard Videos (3 - 10 minutes):** Ideal for medium-depth guides, tutorials, or explanations that cover a topic with moderate complexity.
- **In-Depth Videos (10+ minutes):** Reserved for comprehensive tutorials, case studies, or detailed step-by-step walkthroughs of advanced topics.
- **Level of Detail:**  
The content should thoroughly cover the topic with:
  - **Basic Content (Low Detail):** Simple, brief explanations, usually introductory content with minimal technical depth.
  - **Intermediate Content (Medium Detail):** Moderate coverage with examples and step-by-step instructions.
  - **Advanced Content (High Detail):** Comprehensive coverage including advanced concepts, multiple steps, potential edge cases, and real-world applications.
- **Inclusion of Examples / Code Snippets:**  
Contributions should include working code examples, real-world use cases, or illustrations where applicable.
  - **No examples:** Lower score.
  - **One to two relevant examples:** Medium score.
  - **Multiple, detailed examples that cover edge cases or different use cases:** High score.

## 2. Completeness and Structure

- **Formatting and Organization:**  
The content must be well-structured and easy to follow. Contributions should have clear sections (introduction, body, conclusion) and use headings, bullet points, and other organizational tools.
  - **Poorly structured content:** No clear sections or logical flow; difficult to follow.
  - **Adequately structured content:** Clear sections, logical flow, but minor improvements possible.
  - **Excellent structure:** Professional formatting, logical flow, and highly readable.
- **Coverage of Topic:**  
The contribution should fully address the topic. Completeness is judged by whether all relevant subtopics are covered.
  - **Partial Coverage (Low Completeness):** Only basic aspects of the topic are covered, leaving out key details.
  - **Full Coverage (Medium Completeness):** The topic is covered adequately, though there could be more depth or exploration.
  - **Exhaustive Coverage (High Completeness):** Comprehensive and deep dive into the topic, covering all key aspects thoroughly.

- **Inclusion of Visuals:**

Visual aids such as diagrams, charts, screenshots, or video demonstrations significantly enhance the clarity of the content.

- **No visuals:** Lower score.
- **Some visuals (1-2 images):** Medium score.
- **Multiple, high-quality visuals:** High score.

### 3. Technical Accuracy and Relevance

- **Technical Correctness:**

Content should be factually accurate and align with Rootstock's latest updates and practices. Incorrect or outdated information will receive penalties.

- **Low accuracy:** Contains technical errors or outdated information. (If there are multiple errors the content might not be accepted)
- **Medium accuracy:** Accurate, but with minor technical issues or unclear explanations.
- **High accuracy:** Thoroughly fact-checked and accurate, aligns with the current state of the Rootstock ecosystem.

- **Current and Relevant to the Ecosystem:**

Contributions that cover the latest features, tools, or trends in the Rootstock ecosystem will be rated higher. Contributions that focus on outdated practices or irrelevant content will be rated lower.

- **Low relevance:** Covers obsolete or irrelevant features.
- **Medium relevance:** Useful, but not the most current or impactful topic.
- **High relevance:** Covers new or highly relevant topics that address current needs of the developer community.

### 4. Educational Value and Developer Utility

- **Clarity of Explanation:**

The explanation should be clear, concise, and easy to understand, even for complex topics. Contributions that simplify difficult concepts are highly valued.

- **Poor clarity:** Vague, hard to understand, or convoluted explanations.
- **Moderate clarity:** Understandable but with some unclear sections.
- **High clarity:** Clear, concise, and easy to follow, even for more advanced topics.

- **Practicality and Actionability:**

Content should provide actionable insights, guiding developers through real tasks they can implement. Contributions that are theoretical without practical application will be rated lower.

- **Low practicality:** Theoretical content without clear application.
- **Medium practicality:** Actionable content but limited to a specific, narrow use case.
- **High practicality:** Broadly applicable and directly actionable, with clear instructions for developers to follow.

- **Problem-Solving Focus:**  
Contributions that address real problems faced by developers in the Rootstock ecosystem will be rated higher.
  - **Low impact:** General content without a clear focus on solving developer challenges.
  - **Medium impact:** Addresses a common problem but lacks depth.
  - **High impact:** Directly solves a common or significant problem faced by developers in the ecosystem.

## 5. Innovation and Contribution to Ecosystem

- **New or Unique Content:**  
Contributions that offer new insights, ideas, or solutions that haven't been extensively covered elsewhere will score higher.
  - **Low innovation:** Repeats existing content without adding anything new.
  - **Medium innovation:** Adds some new insights but covers well-trodden ground.
  - **High innovation:** Presents new ideas, techniques, or applications that advance the understanding of Rootstock development.
- **Ecosystem Development and Scalability:**  
Content that contributes to expanding Rootstock's use cases, features, or integrations is highly valuable. Tutorials or case studies that promote Rootstock's scalability, security, or interoperability will score higher.
  - **Low ecosystem contribution:** Minimal impact on Rootstock's growth or development.
  - **Medium ecosystem contribution:** Adds value to specific areas of Rootstock but doesn't significantly broaden its use cases.
  - **High ecosystem contribution:** Encourages broader adoption of Rootstock or opens up new possibilities (e.g., cross-chain integrations, new tools, or libraries).

## 6. Completeness of Supporting Materials

- **Code Snippets / Project Repositories:**  
For guides or tutorials involving code, the contribution should include working code snippets and, where applicable, links to full repositories or project templates.
  - **No code snippets or resources:** Lower score.
  - **Includes relevant code snippets or resources:** Medium score.
  - **Comprehensive, well-documented code and resources:** High score.
- **Troubleshooting and FAQs:**  
Contributions that include a section on common issues, errors, or FAQs provide more value to developers.
  - **No troubleshooting section:** Lower score.
  - **Basic troubleshooting or FAQs:** Medium score.
  - **Comprehensive troubleshooting section or FAQ addressing edge cases:** High score.

---

# Rewards Criteria

## Complexity of Content:

- Higher complexity contributions (e.g., detailed tutorials or advanced case studies) will receive higher rewards than lower complexity contributions (e.g., introductory post).

## Depth and Coverage:

- Contributions that provide in-depth, detailed coverage of the topic, including multiple examples, visuals, and extensive documentation, will be rewarded more generously.

## Educational Value:

- Content that clearly educates and guides developers, helping them solve real-world problems or better understand advanced concepts, will receive higher rewards.

## Innovation and Ecosystem Impact:

- Contributions that introduce new tools, integrations, or techniques to the Rootstock ecosystem, or significantly contribute to its scalability and development, will be highly rewarded.

## Clarity, Structure, and Completeness:

- Contributions that are well-structured, clear, and thorough, with high-quality code examples, visuals, and supporting materials (such as FAQs or troubleshooting), will receive higher rewards.

---

# Guidelines for Technical Documentation

1. See [Contributing to Rootstock Documentation](#) and [Contributor Guidelines](#)
2. Follow the [Rootstock Style Guide](#)

---

# Guidelines Written Content

## 1. Keyword Research and Usage

- **Identify Primary and Secondary Keywords:**  
Use tools like Google Keyword Planner, Ahrefs, or SEMrush to find relevant keywords related to Rootstock and blockchain development. Focus on terms developers are likely to search for, such as “smart contract development,” “dApp integration,” “DeFi on Bitcoin,” etc.
- **Keyword Placement:**
  - Include the **primary keyword** in the following key areas:
    - Title (within the first 60 characters).
    - The first 100 words of the article.
    - At least one subheading (H2 or H3).
    - Naturally throughout the content, ensuring a **keyword density** of around 1-2% (avoid keyword stuffing).
  - Use **secondary keywords** naturally in subheadings, bullet points, and body text to support the main topic.
- **LSI (Latent Semantic Indexing) Keywords:**  
These are related terms and phrases that search engines associate with your primary topic. Incorporate them naturally. For example, for “smart contract development,” you could use LSI terms like “blockchain,” “Ethereum smart contracts,” “dApp testing tools,” etc.

## 2. Title and Meta Description

- **SEO-Friendly Title:**  
Craft a title that is both compelling and optimized for search engines. Keep the title under 60 characters and ensure it includes the primary keyword. Example: “How to Deploy Smart Contracts on Rootstock: Step-by-Step Guide.”
- **Meta Description (150-160 characters):**  
Write a concise summary of the article, ensuring it includes the primary keyword and encourages users to click. Example: “Learn how to write, deploy, and test smart contracts on Rootstock in this detailed step-by-step guide for developers.”

## 3. Headings and Subheadings (H1, H2, H3)

- Use **H1** for the main title (this should contain the primary keyword).
- Break the content into sections using **H2** and **H3** subheadings. Ensure at least one H2 contains the primary keyword, and other subheadings contain variations or related keywords.
- Subheadings should be clear and descriptive to enhance readability and guide users (and search engines) through the article.

## 4. Content Formatting and Readability

- **Short Paragraphs:**  
Keep paragraphs concise (2-4 sentences) to enhance readability, especially on mobile devices.

- **Bullet Points and Numbered Lists:**  
Use bullet points or numbered lists where applicable to break down complex ideas or steps. This helps improve the user experience and makes content more scannable.
- **Use of Bold and Italics:**  
Highlight key points using bold or italics to emphasize important information. However, do not overuse these elements, as it can make the text look cluttered.

## 5. Internal and External Linking

- **Internal Links:**  
Link to relevant Rootstock content, such as previous blog posts, guides, or tutorials that provide further information. Aim for 2-3 internal links per article.
- **External Links:**  
Include 1-2 external links to reputable sources or tools that back up your points (e.g., linking to GitHub repositories, research papers, or widely recognized blockchain platforms).
- **Anchor Text:**  
Use descriptive anchor text that includes keywords or describes the linked content. Avoid generic phrases like "click here."

## 6. Multimedia and Alt Text

- **Images and Diagrams:**  
Include relevant images (e.g., screenshots, diagrams) to support the content. Use descriptive file names for images (e.g., "Rootstock-Smart-Contract-Example.png").
- **Alt Text:**  
Write concise alt text for each image that describes its content and includes the keyword where appropriate. This improves accessibility and helps search engines index the image properly.
- **Videos (if applicable):**  
If embedding videos, include a short description below the video explaining what it covers, along with keywords.

## 7. URL Structure

- Ensure that the URL is short, descriptive, and includes the primary keyword. Example:  
`/deploy-smart-contracts-rootstock-guide`
- Avoid using numbers, symbols, or irrelevant words in the URL.

## 8. Mobile Optimization

- Ensure content is mobile-friendly by using responsive design. This includes breaking up large paragraphs, avoiding too many large images, and ensuring readability on small screens.

## 9. Content Length

- **Minimum word count:** Aim for at least 800 words for blog posts and tutorials. For in-depth guides, target 1,500+ words.
- Longer content generally performs better on search engines if it provides value and is well-structured.

## 10. Proofreading and Grammar

- Check for spelling, grammar, and punctuation errors. Use tools like Grammarly or Hemingway App to ensure readability and a professional tone.
- 

# Guidelines for Video Content

## 1. Video Title and Keywords

- **SEO-Friendly Video Title:**  
Ensure the video title is optimized for search by including relevant keywords. Keep it concise and engaging. Example: "How to Write and Deploy Smart Contracts on Rootstock."
- **Primary and Secondary Keywords:**  
Identify a primary keyword for the video (e.g., "Rootstock smart contracts") and a few secondary keywords related to the topic (e.g., "Bitcoin smart contracts," "DeFi on Rootstock").

## 2. Video Description

- Write a **300-400 word description** for the video that summarizes the content and includes the primary keyword within the first 150 characters. Use secondary keywords throughout the description naturally.
- Include links to related articles, tutorials, or resources (both on the Rootstock platform and external tools).
- Example Description:  
"In this video, learn how to write and deploy smart contracts on the Rootstock blockchain. We'll cover the basics of smart contract development, using tools like Remix and Hardhat. Watch till the end for tips on troubleshooting deployment issues. Read our full guide [link to article]."

## 3. Tags and Metadata

- Use 5-10 relevant tags that include primary and secondary keywords. Example tags: "Rootstock," "smart contracts," "Bitcoin DeFi," "dApp development."
- Ensure metadata (such as the video category) is aligned with blockchain, technology, or software development categories.



#### 4. Transcripts and Captions

- **Upload a transcript** for each video. A transcript improves accessibility and SEO by providing a text version of the content that search engines can index.
- Ensure **captions are accurate** to help with accessibility and improve search ranking, as YouTube and other platforms use captions for indexing.

#### 5. Thumbnail Optimization

- Create a **custom thumbnail** with clear, relevant text and an image that represents the video content. Use colors and designs that grab attention without being overly cluttered.
- Use an **image filename** that includes the keyword. Example:  
`rootstock-smart-contract-tutorial-thumbnail.png`

#### 6. Video Length and Engagement

- **Optimal Video Length:**  
Keep the video length between 5-10 minutes for tutorials or introductory videos. Longer videos (10-20 minutes) are suitable for in-depth technical content.
- **Engagement Tactics:**  
Encourage viewers to engage by including a call to action (CTA), such as "Subscribe for more Rootstock tutorials" or "Check the description for links to related guides."

#### 7. Internal Linking and Cards

- Use **YouTube Cards** or equivalent features on other platforms to link to related videos or playlists during the video. This improves session time and keeps viewers engaged with Rootstock content.

#### 8. Video File Name

- Before uploading, rename the video file to something descriptive and keyword-rich.  
Example: `rootstock-smart-contract-tutorial.mp4`

#### 9. End Screens and CTAs

- Add **end screens** with links to related content (both videos and articles) to encourage users to explore further.

---

### Final Review Process

- Before submitting, ensure the content is **reviewed** for SEO compliance by using tools like **Yoast SEO (for written content)** or **TubeBuddy/vidIQ (for video content)** to check if all SEO elements (keywords, meta tags, etc.) are in place.