Redmine - Feature #35037

Make wiki text section extraction less fragile

2021-04-05 16:00 - Martin Cizek

Priority: Normal Due date: Assignee: % Done: 0% Category: Text formatting Estimated time: 0.00 hour Target version: Resolution: 0.00 hour Description The current approach to per-section text editing is inherently fragile: • Section slinks are generated based on rendered HTML headings (before macro injections). • Section or the markup source is based on simplified regexp-based parsing of the markup, independent on the fully featured markup parser. The issue is going to be more significant after relaxing restrictions on markup syntax (#32424, #35035). But as #35363 shows, even the current restrictions do not guarantee correct section extraction. See also the exmple below. I can imagine two approaches to the solution: 1. For CommonMark - using the sourcepos feature of the renderer. Also applicable to other eventual formatters with similar feature: 2. For any markup - introducing validation of the section extraction results, which would detect that the extraction fails and would disable per-section edit links. I can offer creating a PoC of the sourcepos approach for CommonMark format after #32424 is incorporated. A difficult-to-solve example of Markdown with broken section extraction follows (copied from the skipped unit test in #35035): # Title ## Heading 2 - item one a	Status:	New	Start date:		
Assignee: Assignee: Category: Text formatting Target version: Resolution: Description The current approach to per-section text editing is inherently fragile: - Sections links are generated based on rendered HTML headings (before macro injections). - Section struction from the markup source is based on simplified regexp-based parsing of the markup, independent on the fully featured markup parser. The issue is going to be more significant after relaxing restrictions on markup syntax (#32424, #35035). But as #35036 shows, even the current restrictions do not guarantee correct section extraction. See also the exmple below. I can imagine two approaches to the solution: 1. For CommonMark - using the sourcepos feature of the renderer. Also applicable to other eventual formatters with similar feature. 2. For any markup - introducing validation of the section extraction results, which would detect that the extraction fails and would disable per-section edit links. I can offer creating a PoC of the sourcepos approach for CommonMark format after #32424 is incorporated. A difficult-to-solve example of Markdown with broken section extraction follows (copied from the skipped unit test in #35036): # Title ## Title ## feading 2 - item one 	Priority:	Normal	Due date:		
Category: Text formatting Festimated time: 0.00 hour Target version: Resolution: Description The current approach to per-section text editing is inherently fragile: • Section sinks are generated based on rendered HTML headings (before macro injections). • Section extraction from the markup source is based on simplified regexp-based parsing of the markup, independent on the fully featured markup parser. The issue is going to be more significant after relaxing restrictions on markup syntax (#32424, #35035). But as #35036 shows, even the current restrictions on on guarantee correct section extraction. See also the exmple below. I can imagine two approaches to the solution: 1. For CommonMark - using the sourcepos feature of the renderer. Also applicable to other eventual formatters with similar feature. 1. For any markup - introducing validation of the section extraction results, which would detect that the extraction fails and would disable per-section edit links. I can offer creating a PoC of the sourcepos approach for CommonMark format after #32424 is incorporated. A difficult-to-solve example of Markdown with broken section extraction follows (copied from the skipped unit test in #35036): # Title ## feading 2 - item one - - Related to Redmine - Defect #35036; Markdown text sections br	Assignee:		% Done:	0%	
Category: Text for making Coor hour Target version: Resolution: Description The current approach to per-section text editing is inherently fragile: • Sections links are generated based on rendered HTML headings (before macro injections). • Section straction from the markup source is based on simplified regexp-based parsing of the markup, independent on the fully featured markup parser. The issue is going to be more significant after relaxing restrictions on markup syntax (#32424, #35035). But as #35036 shows, even the current restrictions do not guarantee correct section extraction. See also the exmple below. I can imagine two approaches to the solution: 1. For CommonMark - using the sourcepos feature of the renderer. Also applicable to other eventual formatters with similar feature. 2. For any markup - introducing validation of the section extraction results, which would detect that the extraction fails and would disable per-section edit links. I can offer creating a PoC of the sourcepos approach for CommonMark format after #32424 is incorporated. A difficult-to-solve example of Markdown with broken section extraction follows (copied from the skipped unit test in #35036): # Title ## Heading 2 - item One - Related to Redmine - Defect #35036: Markdown test sections broken by thematic Closed	Category:	Text formatting	Estimated time:		
Parget Version: Pesolution: Description The current approach to per-section text editing is inherently fragile: • Sections links are generated based on rendered HTML headings (before macro injections). • Section extraction from the markup source is based on simplified regexp-based parsing of the markup, independent on the fully featured markup parser. The issue is going to be more significant after relaxing restrictions on markup syntax (#32424, #35035). But as #35036 shows, even the current restrictions do not guarantee correct section extraction. See also the exmple below. I can imagine two approaches to the solution: 1. For CommonMark - using the sourcepos feature of the renderer. Also applicable to other eventual formatters with similar feature. 2. For any markup - introducing validation of the section extraction results, which would detect that the extraction fails and would disable per-section edit links. I can offer creating a PoC of the sourcepos approach for CommonMark format after #32424 is incorporated. A difficult-to-solve example of Markdown with broken section extraction follows (copied from the skipped unit test in #35036): # Title ## Heading 2 - item mot a heading ## Reading 2 Nulla nunc nisi, egestas in ornare vel, posuere ac libero. Pleated issues: Related to Redmine - Defect #35036: Markdown tes	Target version.	Text formatting	Lotinated time.	0.00 1001	
Hesolution: Description The current approach to per-section text editing is inherently fragile: • Sections links are generated based on rendered HTML headings (before macro injections). • Section extraction from the markup source is based on simplified regexp-based parsing of the markup, independent on the fully featured markup parser. The issue is going to be more significant after relaxing restrictions on markup syntax (#32424, #35035). But as #35036 shows, even the current restrictions do not guarantee correct section extraction. See also the exmple below. I can imagine two approaches to the solution: 1. For CommonMark - using the sourcepos feature of the renderer. Also applicable to other eventual formatters with similar feature. 2. For any markup - introducing validation of the section extraction results, which would detect that the extraction fails and would disable per-section edit links. I can offer creating a PoC of the sourcepos approach for CommonMark format after #32424 is incorporated. A difficult-to-solve example of Markdown with broken section extraction follows (copied from the skipped unit test in #35036): # Title ### Heading 2 - item one not a heading ## Heading 2 Nulla nunc nisi, egestas in ornare vel, posuere ac libero. Pelated issues: Related to Redmine - Defect #35036: Markdown text sections	Target version:				
Description The current approach to per-section text editing is inherently fragile: • Sections links are generated based on rendered HTML headings (before macro injections). • Section struction from the markup source is based on simplified regexp-based parsing of the markup, independent on the fully featured markup parser. The issue is going to be more significant after relaxing restrictions on markup syntax (#32424, #35035). But as #35036 shows, even the current restrictions do not guarantee correct section extraction. See also the exmple below. I can imagine two approaches to the solution: 1. For CommonMark - using the sourcepos feature of the renderer. Also applicable to other eventual formatters with similar feature. 2. For any markup - introducing validation of the section extraction results, which would detect that the extraction fails and would disable per-section edit links. I can offer creating a PoC of the sourcepos approach for CommonMark format after #32424 is incorporated. A difficult-to-solve example of Markdown with broken section extraction follows (copied from the skipped unit test in #35036): # Title ## Heading 2 - item 0re - mot a heading ## Heading 2 Wulla nunc nisi, egestas in ornare vel, posuere ac libero. Related to Redmine - Defect #35036: Markdown text sections broken by thematic Closed	Resolution:				
 The current approach to per-section text editing is innerently tragile: Sections links are generated based on rendered HTML headings (before macro injections). Section extraction from the markup source is based on simplified regexp-based parsing of the markup, independent on the fully featured markup parser. The issue is going to be more significant after relaxing restrictions on markup syntax (#32424, #35035). But as #35036 shows, even the current restrictions do not guarantee correct section extraction. See also the exmple below. I can imagine two approaches to the solution: For CommonMark - using the sourcepos feature of the renderer. Also applicable to other eventual formatters with similar feature. For any markup - introducing validation of the section extraction results, which would detect that the extraction fails and would disable per-section edit links. I can offer creating a PoC of the sourcepos approach for CommonMark format after #32424 is incorporated. A difficult-to-solve example of Markdown with broken section extraction follows (copied from the skipped unit test in #35036): # Title # # Heading 2 item one not a heading # Heading 2 Wulla nunc nisi, egestas in ornare vel, posuere ac libero. Related to Redmine - Defect #35036: Markdown text sections broken by thematic Closed 					
 Sections links are generated based on rendered HTML headings (before macro injections). Section extraction from the markup source is based on simplified regexp-based parsing of the markup, independent on the fully featured markup parser. The issue is going to be more significant after relaxing restrictions on markup syntax (#32424, #35035). But as #35036 shows, even the current restrictions do not guarantee correct section extraction. See also the exmple below. I can imagine two approaches to the solution: For CommonMark - using the sourcepos feature of the renderer. Also applicable to other eventual formatters with similar feature. For any markup - introducing validation of the section extraction results, which would detect that the extraction fails and would disable per-section edit links. I can offer creating a PoC of the sourcepos approach for CommonMark format after #32424 is incorporated. A difficult-to-solve example of Markdown with broken section extraction follows (copied from the skipped unit test in #35036): # Title # Heading 2 item one not a heading ## Heading 2 Wulla nunc nisi, egestas in ornare vel, posuere ac libero. Related to Redmine - Defect #35036: Markdown text sections broken by thematic Closed 	The current approach to per-section text editing is inherently tragile:				
The issue is going to be more significant after relaxing restrictions on markup syntax (#32424, #35035). But as #35036 shows, even the current restrictions do not guarantee correct section extraction. See also the exmple below. I can imagine two approaches to the solution: 1. For CommonMark - using the sourcepos feature of the renderer. Also applicable to other eventual formatters with similar feature. 2. For any markup - introducing validation of the section extraction results, which would detect that the extraction fails and would disable per-section edit links. I can offer creating a PoC of the sourcepos approach for CommonMark format after #32424 is incorporated. A difficult-to-solve example of Markdown with broken section extraction follows (copied from the skipped unit test in #35036): # Title ## Heading 2 - item one - not a heading ## Heading 2 Related to Redmine - Defect #35036: Markdown text sections broken by thematic Related to Redmine - Defect #35036: Markdown text sections broken by thematic	 Sections links are generated based on rendered HTML headings (before macro injections). Section extraction from the markup source is based on simplified regexp-based parsing of the markup, independent on the fully featured markup parser. 				
 I can imagine two approaches to the solution: 1. For CommonMark - using the sourcepos feature of the renderer. Also applicable to other eventual formatters with similar feature. 2. For any markup - introducing validation of the section extraction results, which would detect that the extraction fails and would disable per-section edit links. I can offer creating a PoC of the sourcepos approach for CommonMark format after #32424 is incorporated. A difficult-to-solve example of Markdown with broken section extraction follows (copied from the skipped unit test in #35036): # Title ## Heading 2 item one not a heading ## Heading 2 Nulla nunc nisi, egestas in ornare vel, posuere ac libero. Flated issues: Related to Redmine - Defect #35036: Markdown text sections broken by thematic Closed	The issue is going to be more significant after relaxing restrictions on markup syntax (<u>#32424</u> , <u>#35035</u>). But as <u>#35036</u> shows, even the current restrictions do not guarantee correct section extraction. See also the exmple below.				
 1. For CommonMark - using the sourcepos feature of the renderer. Also applicable to other eventual formatters with similar feature. 2. For any markup - introducing validation of the section extraction results, which would detect that the extraction fails and would disable per-section edit links. I can offer creating a PoC of the sourcepos approach for CommonMark format after #32424 is incorporated. A difficult-to-solve example of Markdown with broken section extraction follows (copied from the skipped unit test in #35036): # Title ## Heading 2 item one not a heading ## Heading 2 Nulla nunc nisi, egestas in ornare vel, posuere ac libero. Related to Redmine - Defect #35036: Markdown text sections broken by thematic Closed	I can imagine two approaches to the solution:				
<pre>line of the source of the</pre>	 For CommonMark - using the sourcepos feature of the renderer. Also applicable to other eventual formatters with similar feature. For any markup - introducing validation of the section extraction results, which would detect that the extraction fails and would 				
I can offer creating a PoC of the sourcepos approach for CommonMark format after <u>#32424</u> is incorporated. A difficult-to-solve example of Markdown with broken section extraction follows (copied from the skipped unit test in <u>#35036</u>): # Title # Heading 2 - item one - not a heading ## Heading 2 Nulla nunc nisi, egestas in ornare vel, posuere ac libero. Felated issues: Related to Redmine - Defect #35036: Markdown text sections broken by thematic Closed	disable per-section edit links.				
A difficult-to-solve example of Markdown with broken section extraction follows (copied from the skipped unit test in #35036): # Title ## Heading 2 - item one - not a heading ## Heading 2 Nulla nunc nisi, egestas in ornare vel, posuere ac libero. Related issues: Related to Redmine - Defect #35036: Markdown text sections broken by thematic Closed	I can offer creating a PoC of the sourcepos approach for CommonMark format after <u>#32424</u> is incorporated.				
<pre># Title ## Heading 2 - item one - not a heading ## Heading 2 Nulla nunc nisi, egestas in ornare vel, posuere ac libero. Related issues: Related to Redmine - Defect #35036: Markdown text sections broken by thematic Closed</pre>	A difficult-to-solve example of Markdown with broken section extraction follows (copied from the skipped unit test in #35036):				
<pre>## Heading 2 - item one - not a heading ## Heading 2 Nulla nunc nisi, egestas in ornare vel, posuere ac libero. Related issues: Related to Redmine - Defect #35036: Markdown text sections broken by thematic Closed</pre>	# Title				
- item one - not a heading ## Heading 2 Nulla nunc nisi, egestas in ornare vel, posuere ac libero. Related issues: Related to Redmine - Defect #35036: Markdown text sections broken by thematic Closed	## Heading 2				
one not a heading ## Heading 2 Nulla nunc nisi, egestas in ornare vel, posuere ac libero. Related issues: Related to Redmine - Defect #35036: Markdown text sections broken by thematic Closed	- item				
not a heading ## Heading 2 Nulla nunc nisi, egestas in ornare vel, posuere ac libero. Related issues: Related to Redmine - Defect #35036: Markdown text sections broken by thematic Closed	one				
<pre>## Heading 2 Nulla nunc nisi, egestas in ornare vel, posuere ac libero. Related issues: Related to Redmine - Defect #35036: Markdown text sections broken by thematic Closed</pre>	not a heading				
Nulla nunc nisi, egestas in ornare vel, posuere ac libero. Related issues: Related to Redmine - Defect #35036: Markdown text sections broken by thematic Closed	## Heading 2				
Related issues: Closed Related to Redmine - Defect #35036: Markdown text sections broken by thematic Closed	Nulla nunc nisi, egestas in ornare vel, posuere ac libero.				
Related to Redmine - Defect #35036: Markdown text sections broken by thematic Closed	Related issues:				

History

#1 - 2021-05-12 06:48 - Go MAEDA

- Related to Defect #35036: Markdown text sections broken by thematic breaks (horizontal rules) added