

# Guide d'utilisation de AIT

REVISION HISTORY
------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>Installation</b>	<b>1</b>
1.1	Avec PEAR . . . . .	1
1.2	sans PEAR . . . . .	1
<b>2</b>	<b>Modèle de données</b>	<b>2</b>
<b>3</b>	<b>Constantes pré-définies</b>	<b>3</b>
<b>4</b>	<b>Connexion/Configuration à Base de Données</b>	<b>4</b>
4.1	SGBD supportée . . . . .	4
4.1.1	Pré-requis . . . . .	4
4.1.2	Connexion à la base de données . . . . .	4
4.1.2.1	Synopsis . . . . .	4
4.1.2.2	Description . . . . .	4
4.1.2.3	Paramètres . . . . .	4
4.1.2.4	Retour . . . . .	5
4.1.2.5	Exemple . . . . .	5
4.1.3	Création et contrôle de la structure de données . . . . .	5
<b>5</b>	<b>Les erreurs et leur gestion</b>	<b>6</b>
<b>6</b>	<b>Debugger</b>	<b>7</b>
<b>7</b>	<b>Développement</b>	<b>8</b>
7.1	Modification & Ajout . . . . .	8
7.2	Tests Unitaires . . . . .	8
<b>8</b>	<b>Références des classes</b>	<b>9</b>
8.1	AITPDO . . . . .	9
8.2	AITSchema . . . . .	9
8.2.1	__construct . . . . .	9
8.2.1.1	Synopsis . . . . .	9
8.2.1.2	Description . . . . .	9

---

8.2.1.3	Paramètres	9
8.2.2	__get	10
8.2.2.1	Synopsis	10
8.2.2.2	Description	10
8.2.2.3	Paramètres	10
8.2.2.4	Retour	10
8.3	AITQuery	10
8.3.1	__construct	10
8.3.1.1	Synopsis	10
8.3.1.2	Description	10
8.3.1.3	Paramètres	10
8.3.2	all	10
8.3.2.1	Synopsis	10
8.3.2.2	Description	11
8.3.2.3	Paramètres	11
8.3.2.4	Retour	11
8.3.3	one	11
8.3.3.1	Synopsis	11
8.3.3.2	Description	11
8.3.3.3	Paramètres	11
8.3.3.4	Retour	11
8.3.4	getSQL	11
8.3.4.1	Synopsis	11
8.3.4.2	Description	11
8.3.4.3	Paramètres	11
8.3.4.4	Retour	12
8.4	AIT_ItemType	12
8.4.1	__construct	12
8.4.1.1	Synopsis	12
8.4.1.2	Description	12
8.4.1.3	Paramètres	12
8.4.2	defTagType	12
8.4.2.1	Synopsis	12
8.4.2.2	Description	12
8.4.2.3	Paramètres	12
8.4.2.4	Retour	12
8.4.3	addTagType	13
8.4.3.1	Synopsis	13
8.4.3.2	Description	13

---

---

8.4.3.3	Paramètres	13
8.4.3.4	Retour	13
8.4.4	getTagType	13
8.4.4.1	Synopsis	13
8.4.4.2	Description	13
8.4.4.3	Paramètres	13
8.4.4.4	Retour	13
8.4.5	newItem	13
8.4.5.1	Synopsis	13
8.4.5.2	Description	13
8.4.5.3	Paramètres	14
8.4.5.4	Retour	14
8.4.6	addItem	14
8.4.6.1	Synopsis	14
8.4.6.2	Description	14
8.4.6.3	Paramètres	14
8.4.6.4	Retour	14
8.4.7	getItem	14
8.4.7.1	Synopsis	14
8.4.7.2	Description	14
8.4.7.3	Paramètres	14
8.4.7.4	Retour	14
8.4.7.5	Exemple	15
8.4.8	defItem	15
8.4.8.1	Synopsis	15
8.4.8.2	Description	15
8.4.8.3	Paramètres	15
8.4.8.4	Retour	15
8.4.9	getItems	15
8.4.9.1	Synopsis	15
8.4.9.2	Description	15
8.4.9.3	Paramètres	15
8.4.9.4	Retour	16
8.4.9.5	Exemple	16
8.4.10	fetchItems	16
8.4.10.1	Synopsis	16
8.4.10.2	Description	16
8.4.10.3	Paramètres	16
8.4.10.4	Retour	16

---

---

8.4.10.5	Exemple	17
8.4.11	getTagTypes	17
8.4.11.1	Synopsis	17
8.4.11.2	Description	17
8.4.11.3	Paramètres	17
8.4.11.4	Retour	17
8.4.12	searchItems	17
8.4.12.1	Synopsis	17
8.4.12.2	Description	18
8.4.12.3	Paramètres	18
8.4.12.4	Retour	18
8.4.13	getItemBySystemID	18
8.4.13.1	Synopsis	18
8.4.13.2	Description	18
8.4.13.3	Paramètres	18
8.4.13.4	Retour	18
8.4.13.5	Exemple	18
8.4.14	queryItems	19
8.4.14.1	Synopsis	19
8.4.14.2	Description	19
8.4.14.3	Paramètres	19
8.4.14.4	Retour	19
8.4.14.5	Exemple	19
8.4.15	countItems	20
8.4.15.1	Synopsis	20
8.4.15.2	Description	20
8.4.15.3	Paramètres	20
8.4.15.4	Retour	20
8.4.15.5	Exemple	20
8.4.16	countTagTypes	20
8.4.16.1	Synopsis	20
8.4.16.2	Description	20
8.4.16.3	Paramètres	20
8.4.16.4	Retour	21
8.4.17	del	21
8.4.17.1	Synopsis	21
8.4.17.2	Description	21
8.4.17.3	Paramètres	21
8.4.17.4	Retour	21

---

---

8.4.18	getPDO	21
8.4.18.1	Synopsis	21
8.4.18.2	Description	21
8.4.18.3	Paramètres	21
8.4.18.4	Retour	21
8.4.19	ren	21
8.4.19.1	Synopsis	21
8.4.19.2	Description	21
8.4.19.3	Paramètres	21
8.4.19.4	Retour	22
8.4.20	exists	22
8.4.20.1	Synopsis	22
8.4.20.2	Description	22
8.4.20.3	Paramètres	22
8.4.20.4	Retour	22
8.4.21	getSystemID	22
8.4.21.1	Synopsis	22
8.4.21.2	Description	22
8.4.21.3	Paramètres	22
8.4.21.4	Retour	22
8.4.22	get	23
8.4.22.1	Synopsis	23
8.4.22.2	Description	23
8.4.22.3	Paramètres	23
8.4.22.4	Retour	23
8.4.23	set	23
8.4.23.1	Synopsis	23
8.4.23.2	Description	23
8.4.23.3	Paramètres	23
8.4.23.4	Retour	23
8.4.24	search	23
8.4.24.1	Synopsis	23
8.4.24.2	Description	24
8.4.24.3	Paramètres	24
8.4.24.4	Retour	24
8.4.25	getTimestamps	24
8.4.25.1	Synopsis	24
8.4.25.2	Description	24
8.4.25.3	Paramètres	24

---

---

8.4.25.4	Retour	24
8.4.25.5	Exemple	24
8.4.26	getScore	25
8.4.26.1	Synopsis	25
8.4.26.2	Description	25
8.4.26.3	Paramètres	25
8.4.26.4	Retour	25
8.4.27	setScore	25
8.4.27.1	Synopsis	25
8.4.27.2	Description	25
8.4.27.3	Paramètres	25
8.4.27.4	Retour	26
8.4.28	setFrequency	26
8.4.28.1	Synopsis	26
8.4.28.2	Description	26
8.4.28.3	Paramètres	26
8.4.28.4	Retour	26
8.4.29	debug	26
8.4.29.1	Synopsis	26
8.4.29.2	Description	26
8.4.29.3	Paramètres	26
8.4.29.4	Retour	26
8.4.30	dump	26
8.4.30.1	Synopsis	26
8.4.30.2	Description	27
8.4.30.3	Paramètres	27
8.4.30.4	Retour	27
8.4.31	getBySystemID	27
8.4.31.1	Synopsis	27
8.4.31.2	Description	27
8.4.31.3	Paramètres	27
8.4.31.4	Retour	27
8.4.31.5	Exemple	27
8.5	AIT_Item	28
8.5.1	__construct	28
8.5.1.1	Synopsis	28
8.5.1.2	Description	28
8.5.1.3	Paramètres	28
8.5.1.4	Synopsis	29

---



---

8.5.1.5	Description	29
8.5.1.6	Paramètres	29
8.5.1.7	Retour	29
8.5.2	addTag	29
8.5.2.1	Synopsis	29
8.5.2.2	Description	29
8.5.2.3	Paramètres	29
8.5.2.4	Retour	30
8.5.3	delTag	30
8.5.3.1	Synopsis	30
8.5.3.2	Description	30
8.5.3.3	Paramètres	30
8.5.3.4	Retour	30
8.5.4	attach	30
8.5.4.1	Synopsis	30
8.5.4.2	Description	30
8.5.4.3	Paramètres	30
8.5.4.4	Retour	31
8.5.5	detach	31
8.5.5.1	Synopsis	31
8.5.5.2	Description	31
8.5.5.3	Paramètres	31
8.5.5.4	Retour	31
8.5.6	getTags	31
8.5.6.1	Synopsis	31
8.5.6.2	Description	31
8.5.6.3	Paramètres	31
8.5.6.4	Retour	31
8.5.7	getTypedTags	32
8.5.7.1	Synopsis	32
8.5.7.2	Description	32
8.5.7.3	Paramètres	32
8.5.7.4	Retour	32
8.5.7.5	Exemple	32
8.5.8	fetchTags	32
8.5.8.1	Synopsis	32
8.5.8.2	Description	32
8.5.8.3	Paramètres	33
8.5.8.4	Retour	33

---

---

8.5.9	getItems	33
8.5.9.1	Synopsis	33
8.5.9.2	Description	33
8.5.9.3	Paramètres	33
8.5.9.4	Retour	33
8.5.10	getTypedItems	33
8.5.10.1	Synopsis	33
8.5.10.2	Description	34
8.5.10.3	Paramètres	34
8.5.10.4	Retour	34
8.5.11	fetchItems	34
8.5.11.1	Synopsis	34
8.5.11.2	Description	34
8.5.11.3	Paramètres	34
8.5.11.4	Retour	34
8.5.11.5	Synopsis	35
8.5.11.6	Description	35
8.5.11.7	Paramètres	35
8.5.11.8	Retour	35
8.5.12	fetchItems	36
8.5.12.1	Synopsis	36
8.5.12.2	Description	36
8.5.12.3	Paramètres	36
8.5.12.4	Retour	36
8.5.13	del	36
8.5.13.1	Synopsis	36
8.5.13.2	Description	36
8.5.13.3	Paramètres	36
8.5.13.4	Retour	36
8.5.14	getItemType	37
8.5.14.1	Synopsis	37
8.5.14.2	Description	37
8.5.14.3	Paramètres	37
8.5.14.4	Retour	37
8.5.15	getTagsObject	37
8.5.15.1	Synopsis	37
8.5.15.2	Description	37
8.5.15.3	Paramètres	37
8.5.15.4	Retour	37

---

---

8.5.15.5	Exemple	37
8.5.16	getPDO	38
8.5.16.1	Synopsis	38
8.5.16.2	Description	38
8.5.16.3	Paramètres	38
8.5.16.4	Retour	38
8.5.17	ren	38
8.5.17.1	Synopsis	38
8.5.17.2	Description	38
8.5.17.3	Paramètres	38
8.5.17.4	Retour	38
8.5.18	exists	38
8.5.18.1	Synopsis	38
8.5.18.2	Description	38
8.5.18.3	Paramètres	39
8.5.18.4	Retour	39
8.5.19	getSystemID	39
8.5.19.1	Synopsis	39
8.5.19.2	Description	39
8.5.19.3	Paramètres	39
8.5.19.4	Retour	39
8.5.20	get	39
8.5.20.1	Synopsis	39
8.5.20.2	Description	39
8.5.20.3	Paramètres	39
8.5.20.4	Retour	40
8.5.21	set	40
8.5.21.1	Synopsis	40
8.5.21.2	Description	40
8.5.21.3	Paramètres	40
8.5.21.4	Retour	40
8.5.22	search	40
8.5.22.1	Synopsis	40
8.5.22.2	Description	40
8.5.22.3	Paramètres	40
8.5.22.4	Retour	41
8.5.23	getTimestamps	41
8.5.23.1	Synopsis	41
8.5.23.2	Description	41

---

---

8.5.23.3	Paramètres	41
8.5.23.4	Retour	41
8.5.23.5	Exemple	41
8.5.24	getScore	41
8.5.24.1	Synopsis	41
8.5.24.2	Description	42
8.5.24.3	Paramètres	42
8.5.24.4	Retour	42
8.5.25	setScore	42
8.5.25.1	Synopsis	42
8.5.25.2	Description	42
8.5.25.3	Paramètres	42
8.5.25.4	Retour	42
8.5.26	setFrequency	42
8.5.26.1	Synopsis	42
8.5.26.2	Description	42
8.5.26.3	Paramètres	42
8.5.26.4	Retour	43
8.5.27	debug	43
8.5.27.1	Synopsis	43
8.5.27.2	Description	43
8.5.27.3	Paramètres	43
8.5.27.4	Retour	43
8.5.28	dump	43
8.5.28.1	Synopsis	43
8.5.28.2	Description	43
8.5.28.3	Paramètres	43
8.5.28.4	Retour	43
8.5.29	getBySystemID	44
8.5.29.1	Synopsis	44
8.5.29.2	Description	44
8.5.29.3	Paramètres	44
8.5.29.4	Retour	44
8.5.29.5	Exemple	44
8.6	AIT_TagType	44
8.6.1	__construct	44
8.6.1.1	Synopsis	44
8.6.1.2	Description	44
8.6.1.3	Paramètres	45

---

---

8.6.2	newTag	45
8.6.2.1	Synopsis	45
8.6.2.2	Description	45
8.6.2.3	Paramètres	45
8.6.2.4	Retour	45
8.6.2.5	Exemple	45
8.6.2.6	Synopsis	46
8.6.2.7	Description	46
8.6.2.8	Paramètres	46
8.6.2.9	Retour	46
8.6.3	getTag	46
8.6.3.1	Synopsis	46
8.6.3.2	Description	46
8.6.3.3	Paramètres	46
8.6.3.4	Retour	47
8.6.4	getTags	47
8.6.4.1	Synopsis	47
8.6.4.2	Description	47
8.6.4.3	Paramètres	47
8.6.4.4	Retour	47
8.6.5	getTagBySystemID	47
8.6.5.1	Synopsis	47
8.6.5.2	Description	47
8.6.5.3	Paramètres	47
8.6.5.4	Retour	47
8.6.6	countItems	48
8.6.6.1	Synopsis	48
8.6.6.2	Description	48
8.6.6.3	Paramètres	48
8.6.6.4	Retour	48
8.6.7	del	48
8.6.7.1	Synopsis	48
8.6.7.2	Description	48
8.6.7.3	Paramètres	48
8.6.7.4	Retour	48
8.6.8	searchTags	48
8.6.8.1	Synopsis	48
8.6.8.2	Description	49
8.6.8.3	Paramètres	49

---

---

8.6.8.4	Retour	49
8.6.9	getPDO	49
8.6.9.1	Synopsis	49
8.6.9.2	Description	49
8.6.9.3	Paramètres	49
8.6.9.4	Retour	49
8.6.10	ren	49
8.6.10.1	Synopsis	49
8.6.10.2	Description	49
8.6.10.3	Paramètres	50
8.6.10.4	Retour	50
8.6.11	exists	50
8.6.11.1	Synopsis	50
8.6.11.2	Description	50
8.6.11.3	Paramètres	50
8.6.11.4	Retour	50
8.6.12	getSystemID	50
8.6.12.1	Synopsis	50
8.6.12.2	Description	50
8.6.12.3	Paramètres	50
8.6.12.4	Retour	50
8.6.13	get	51
8.6.13.1	Synopsis	51
8.6.13.2	Description	51
8.6.13.3	Paramètres	51
8.6.13.4	Retour	51
8.6.14	set	51
8.6.14.1	Synopsis	51
8.6.14.2	Description	51
8.6.14.3	Paramètres	51
8.6.14.4	Retour	52
8.6.15	search	52
8.6.15.1	Synopsis	52
8.6.15.2	Description	52
8.6.15.3	Paramètres	52
8.6.15.4	Retour	52
8.6.16	getTimestamps	52
8.6.16.1	Synopsis	52
8.6.16.2	Description	52

---

---

8.6.16.3	Paramètres	52
8.6.16.4	Retour	52
8.6.16.5	Exemple	53
8.6.17	getScore	53
8.6.17.1	Synopsis	53
8.6.17.2	Description	53
8.6.17.3	Paramètres	53
8.6.17.4	Retour	53
8.6.18	setScore	53
8.6.18.1	Synopsis	53
8.6.18.2	Description	53
8.6.18.3	Paramètres	54
8.6.18.4	Retour	55
8.6.19	setFrequency	55
8.6.19.1	Synopsis	55
8.6.19.2	Description	55
8.6.19.3	Paramètres	55
8.6.19.4	Retour	55
8.6.20	debug	55
8.6.20.1	Synopsis	55
8.6.20.2	Description	55
8.6.20.3	Paramètres	55
8.6.20.4	Retour	55
8.6.21	dump	55
8.6.21.1	Synopsis	55
8.6.21.2	Description	56
8.6.21.3	Paramètres	56
8.6.21.4	Retour	56
8.6.22	getBySystemID	56
8.6.22.1	Synopsis	56
8.6.22.2	Description	56
8.6.22.3	Paramètres	56
8.6.22.4	Retour	56
8.6.22.5	Exemple	56
8.7	AIT_Tag	57
8.7.1	__construct	57
8.7.1.1	Synopsis	57
8.7.1.2	Description	57
8.7.1.3	Paramètres	57

---

---

8.7.2	detach . . . . .	57
8.7.2.1	Synopsis . . . . .	57
8.7.2.2	Description . . . . .	57
8.7.2.3	Paramètres . . . . .	57
8.7.2.4	Retour . . . . .	58
8.7.2.5	Exemple . . . . .	58
8.7.3	attach . . . . .	58
8.7.3.1	Synopsis . . . . .	58
8.7.3.2	Description . . . . .	58
8.7.3.3	Paramètres . . . . .	58
8.7.3.4	Retour . . . . .	58
8.7.4	getRelatedTags . . . . .	59
8.7.4.1	Synopsis . . . . .	59
8.7.4.2	Description . . . . .	59
8.7.4.3	Paramètres . . . . .	59
8.7.4.4	Retour . . . . .	59
8.7.5	fetchRelatedTags . . . . .	59
8.7.5.1	Synopsis . . . . .	59
8.7.5.2	Description . . . . .	59
8.7.5.3	Paramètres . . . . .	59
8.7.5.4	Retour . . . . .	60
8.7.6	countItems . . . . .	60
8.7.6.1	Synopsis . . . . .	60
8.7.6.2	Description . . . . .	60
8.7.6.3	Paramètres . . . . .	60
8.7.6.4	Retour . . . . .	60
8.7.6.5	Exemple . . . . .	60
8.7.7	del . . . . .	60
8.7.7.1	Synopsis . . . . .	60
8.7.7.2	Description . . . . .	60
8.7.7.3	Paramètres . . . . .	60
8.7.7.4	Retour . . . . .	61
8.7.8	getTagType . . . . .	61
8.7.8.1	Synopsis . . . . .	61
8.7.8.2	Description . . . . .	61
8.7.8.3	Paramètres . . . . .	61
8.7.8.4	Retour . . . . .	61
8.7.9	getItems . . . . .	61
8.7.9.1	Synopsis . . . . .	61

---



---

8.7.9.2	Description	61
8.7.9.3	Paramètres	61
8.7.9.4	Retour	61
8.7.10	getPDO	62
8.7.10.1	Synopsis	62
8.7.10.2	Description	62
8.7.10.3	Paramètres	62
8.7.10.4	Retour	62
8.7.11	ren	62
8.7.11.1	Synopsis	62
8.7.11.2	Description	62
8.7.11.3	Paramètres	62
8.7.11.4	Retour	62
8.7.12	exists	62
8.7.12.1	Synopsis	62
8.7.12.2	Description	62
8.7.12.3	Paramètres	63
8.7.12.4	Retour	63
8.7.13	getSystemID	63
8.7.13.1	Synopsis	63
8.7.13.2	Description	63
8.7.13.3	Paramètres	63
8.7.13.4	Retour	63
8.7.14	get	63
8.7.14.1	Synopsis	63
8.7.14.2	Description	63
8.7.14.3	Paramètres	63
8.7.14.4	Retour	64
8.7.15	set	64
8.7.15.1	Synopsis	64
8.7.15.2	Description	64
8.7.15.3	Paramètres	64
8.7.15.4	Retour	64
8.7.16	search	64
8.7.16.1	Synopsis	64
8.7.16.2	Description	64
8.7.16.3	Paramètres	64
8.7.16.4	Retour	65
8.7.17	getTimestamps	65

---

8.7.17.1	Synopsis	65
8.7.17.2	Description	65
8.7.17.3	Paramètres	65
8.7.17.4	Retour	65
8.7.17.5	Exemple	65
8.7.18	getScore	65
8.7.18.1	Synopsis	65
8.7.18.2	Description	66
8.7.18.3	Paramètres	66
8.7.18.4	Retour	66
8.7.19	setScore	66
8.7.19.1	Synopsis	66
8.7.19.2	Description	66
8.7.19.3	Paramètres	66
8.7.19.4	Retour	66
8.7.20	setFrequency	66
8.7.20.1	Synopsis	66
8.7.20.2	Description	66
8.7.20.3	Paramètres	66
8.7.20.4	Retour	67
8.7.21	debug	67
8.7.21.1	Synopsis	67
8.7.21.2	Description	67
8.7.21.3	Paramètres	67
8.7.21.4	Retour	67
8.7.22	dump	67
8.7.22.1	Synopsis	67
8.7.22.2	Description	67
8.7.22.3	Paramètres	67
8.7.22.4	Retour	67
8.7.23	getBySystemID	68
8.7.23.1	Synopsis	68
8.7.23.2	Description	68
8.7.23.3	Paramètres	68
8.7.23.4	Retour	68
8.7.23.5	Exemple	68
8.8	AIT	68
<b>9</b>	<b>Maintenance</b>	<b>69</b>
9.1	Contrôler la cohérence de la base	69
9.2	Visualiser les tags non utilisés	69
9.3	Visualiser les items non utilisés	69

# Preface

AIT est un ensemble de classes PHP permettant le stockage et la recherche de métadonnées. AIT se base sur 2 concepts : des items et des tags. A partir de là, on peut stocker et effectuer des recherches toutes informations formalisées sur ces 2 concepts.

Concrètement AIT peut être considéré simplement comme un moteur de Tag (Tag's engine) en PHP, ou alors si on pousse le raisonnement à l'extrême, comme un moteur générique de stockage d'information.

---

# Chapter 1

## Installation

AIT se présente sous forme d'un fichier AIT-X.Y.Z.tgz. Ce fichier s'installe de deux manières.

### 1.1 Avec PEAR

Pour installer :

```
$ pear install AIT-X.Y.Z.tgz
```

Pour désinstaller :

```
$ pear uninstall AIT
```

Pour mettre à jour :

```
$ pear upgrade AIT-X.Y.Z.tgz
```

### 1.2 sans PEAR

Il suffit de décompresser l'archive dans un répertoire et d'ajuster la variable PHP `include_path` en fonction du répertoire que vous avez choisi.

## Chapter 2

# Modèle de données

AIT traite 4 concepts qui est important de bien comprendre. A chacun des ces quatres concepts est associé une classe PHP :

- Les types d'items (AIT\_ItemType)
- Les items (AIT\_Item)
- Les types de tags (AIT\_TagType)
- Les tags (AIT\_Tag)

Ces 4 concepts respectent le modèle de données conceptuel suivant :

Modèle de données virtuel

---

**Note**

Le modèle de données physique ne se présente pas de cette manière

---

## Chapter 3

# Constantes pré-définies

Table 3.1: Constantes pour le tri

Constante	Valeur	Description
ORDER_ASC (integer)	2	Demande un tri ascendant
ORDER_DESC (integer)	4	Demande un tri descendant
ORDER_BY_LABEL (integer)	8	Demande un tri sur le label
ORDER_BY_SCORE (integer)	18	Demande un tri sur le score

## Chapter 4

# Connexion/Configuration à Base de Données

### 4.1 SGBD supportée

AIT peut théoriquement fonctionner sur tous les SGBD supportés par l'extension PDO. En pratique AIT n'a été testé qu'avec MySQL.

#### 4.1.1 Pré-requis

AIT a besoin qu'une base de données soit créée peu importe son nom.

Création d'une base de données avec Mysql CREATE DATABASE AIT;

#### 4.1.2 Connexion à la base de données

AIT utilise l'extension **PHP PDO**. Pour se connecter à la base on utilise donc une chaîne de connexion PDO.

La méthode AIT::connect prend les même paramètres que **le constructeur** de l'objet PDO.

##### 4.1.2.1 Synopsis

```
PDO connect(string $dsn, string $username = 'root', string $password= '', array $driver_o
```

##### 4.1.2.2 Description

Méthode statique permettant de se connecter à une base de données.

##### 4.1.2.3 Paramètres

string <b>\$dsn</b>	chaîne de connexion
string <b>\$username</b>	user de connexion
string <b>\$password</b>	mot de passe de connexion

array **\$driver\_options** options pour pdo

array **\$ait\_options** options pour pdo

#### 4.1.2.4 Retour

Un objet AITPDO qui dérive l'objet PDO.

#### 4.1.2.5 Exemple

```
<?php

require_once 'AIT.php';

// Paramètres de connexion
$dsn      = 'mysql:host=localhost;dbname=music';
$user     = 'root';
$password = '';

// Connexion à la base
$db = AIT::connect (
    $dsn,
    $user,
    $password
);

// On purge
$db->exec("TRUNCATE ".$db->tag());
$db->exec("TRUNCATE ".$db->tagged());

// Contrôle et création de la structure de données
$db->checkup();
```



#### Warning

IL n'est pas possible d'utiliser directement un objet PDO. AIT pour fonctionner utilise un objet spécifique dérivé de l'objet PDO.

### 4.1.3 Création et contrôle de la structure de données

AIT contrôle et crée la structure de données nécessaire à son bon fonctionnement de manière automatique à condition de faire appel à la méthode `checkup`.

```
$cnxstr = 'mysql:host=localhost;dbname=music';
$db = AIT::connect($cnxstr, 'root');
$db->checkup();
```



## Chapter 5

# Les erreurs et leur gestion

AIT fait appel à la fonction PHP `trigger_error` à chaque fois qu'une erreur est rencontrée. Le niveau d'erreur est toujours `E_USER_ERROR`. A chaque fois que AIT rencontre une erreur l'exécution du code est donc stoppé.

## Chapter 6

# Debugger

AIT possède un mode **debugging** qui permet notamment d'afficher les requêtes SQL envoyés à la base. Ce mode peut-être activé et désactivé globalement ou ponctuellement. Dans ce dernier cas, seule les requêtes produisent entre le moment ou le mode debug est activé et le moment ou il est désactivé seront affichées.

```
AIT::$debugging = true;  
jazz = $schema->style->getTag('Jazz');  
AIT::$debugging = false;
```

## Chapter 7

# Développement

### 7.1 Modification & Ajout

Le code source de AIT est disponible sur Github.com : <http://github.com/touv/ait/tree/master>

### 7.2 Tests Unitaires

Le développement d'AIT est piloté par les tests unitaires. Pour cela on utilise **PHPUnit**

---

**Example 7.1** Pour tester le code :

```
% phpunit AITTest.php
% phpunit AITExtentedTest.php
```

---

Bien évidemment si l'on ajoute une fonctionnalité celle-ci devra posséder son propre test.

---

## Chapter 8

# Références des classes

Le cœur d'AIT est composé de 4 classes de bases qui hérite d'une seule classe mère.

### 8.1 AITPDO

Cette classe permet de pointer sur la base données que l'on souhaite utiliser.

### 8.2 AITSchema

Cette classe permet de créer et de manipuler des schémas de données.

#### 8.2.1 \_\_construct

##### 8.2.1.1 Synopsis

```
void __construct(PDOAIT $pdo, $name, array $attributs)
```

##### 8.2.1.2 Description

Le constructeur de la classe.

##### 8.2.1.3 Paramètres

AITPDO <b>\$pdo</b>	L'instance de base AIT que l'on souhaite interroger.
string <b>\$name</b>	Nom du schéma
array <b>\$attributs</b>	Liste des éléments du schéma

---

## 8.2.2 `__get`

### 8.2.2.1 Synopsis

```
mixed __get(string $name)
```

### 8.2.2.2 Description

Retroune un objet `AIT_TypeItem` si on donne en argument le nom du schéma. `AIT_TypeTag` si on donne en argument le nom d'un des élément du schéma.

### 8.2.2.3 Paramètres

<code>string \$name</code>	Nom d'un élément du schéma
----------------------------	----------------------------

### 8.2.2.4 Retour

mixed

## 8.3 AITQuery

Cette classe permet de créer des requêtes avancées pour trouver des items.

### 8.3.1 `__construct`

#### 8.3.1.1 Synopsis

```
function __construct(AITPDO $pdo)
```

#### 8.3.1.2 Description

Le constructeur de la classe.

#### 8.3.1.3 Paramètres

<code>AITPDO \$pdo</code>	L'instance de base AIT que l'on souhaite interroger.
---------------------------	--

### 8.3.2 `all`

#### 8.3.2.1 Synopsis

```
public function all(ArrayObject $tags)
```

---

### 8.3.2.2 Description

On recherche des items ayant tous les tags donnés en paramètre.

### 8.3.2.3 Paramètres

<b>\$tags</b>	Contient des objet AIT_Tag représentant les tags que l'on souhaite rechercher
---------------	---

### 8.3.2.4 Retour

Retourne FALSE si aucun tag n'a été trouvé dans le tableau d'entrée sinon retourne TRUE.

## 8.3.3 one

### 8.3.3.1 Synopsis

```
public function one(ArrayObject $tags)
```

### 8.3.3.2 Description

On recherche des items ayant au moins l'un des tags donnés en paramètre.

### 8.3.3.3 Paramètres

<b>\$tags</b>	Contient des objet AIT_Tag représentant les tags que l'on souhaite rechercher
---------------	---

### 8.3.3.4 Retour

Retourne false si aucun tag n'a été trouvé dans le tableau d'entrée sinon retourne true.

## 8.3.4 getSQL

### 8.3.4.1 Synopsis

```
public function getSQL()
```

### 8.3.4.2 Description

Retourne la requête construite avec les méthodes one et all sous forme de requête SQL.

### 8.3.4.3 Paramètres

Aucun.

---

#### 8.3.4.4 Retour

Retourne une chaîne de caractère contenant une requête SQL.

## 8.4 AIT\_ItemType

Cette classe permet de manipuler un TYPE d'ITEM.

### 8.4.1 \_\_construct

#### 8.4.1.1 Synopsis

```
void __construct(string $l, PDOAIT $pdo)
```

#### 8.4.1.2 Description

Constructeur

#### 8.4.1.3 Paramètres

string <b>\$label</b>	Label du type d'item
PDOAIT <b>\$pdo</b>	Instance de base AIT que l'on souhaite utiliser.

### 8.4.2 defTagType

#### 8.4.2.1 Synopsis

```
AIT_TagType defTagType(string $l)
```

#### 8.4.2.2 Description

Retourne un type de tag associé au type d'item courant. Si le Tag n'existe pas il est automatiquement créé.

#### 8.4.2.3 Paramètres

string <b>\$l</b>	label
-------------------	-------

#### 8.4.2.4 Retour

Un objet AIT\_TagType représentant le type de tag

### 8.4.3 addTagType

#### 8.4.3.1 Synopsis

```
AIT_TagType addTagType(string $l)
```

#### 8.4.3.2 Description

Ajout d'un type de tag au type d'item courant

#### 8.4.3.3 Paramètres

string \$l	label
------------	-------

#### 8.4.3.4 Retour

Un objet AIT\_TagType représentant le nouveau type de tag

### 8.4.4 getTagType

#### 8.4.4.1 Synopsis

```
AIT_TagType getTagType(string $l)
```

#### 8.4.4.2 Description

Retourne un type de tag du type d'item courant en fonction de son label.

#### 8.4.4.3 Paramètres

string \$l	Le label.
------------	-----------

#### 8.4.4.4 Retour

Un objet AIT\_TagType représentant le type de tag trouvé ou NULL si il n'existe pas.

### 8.4.5 newItem

#### 8.4.5.1 Synopsis

```
AIT_Item newItem()
```

#### 8.4.5.2 Description

Crée un item. Son label est calculé automatiquement en fonction de l'identifiant système.



### 8.4.5.3 Paramètres

Aucun.

### 8.4.5.4 Retour

Un objet AIT\_Item représentant le nouvel item.

## 8.4.6 addItem

### 8.4.6.1 Synopsis

```
AIT_Item addItem(string $l)
```

### 8.4.6.2 Description

Ajout d'un item au type d'item courant

### 8.4.6.3 Paramètres

string \$l	nom du nouveau item
------------	---------------------

### 8.4.6.4 Retour

Un objet AIT\_Item représentant le nouvel item.

## 8.4.7 getItem

### 8.4.7.1 Synopsis

```
AIT_Item getItem(string $l)
```

### 8.4.7.2 Description

Récupère un item

### 8.4.7.3 Paramètres

string \$l	nom de l'item
------------	---------------

### 8.4.7.4 Retour

Un objet AIT\_Item représentant l'item trouvé.

---

### 8.4.7.5 Exemple

```
<?php
require_once 'prepend.php';

$item = $schema->disques->getItem('a1902425');
if (!is_null($item)) {
    echo $item->get();
}
// Affichera : a1902425
```

### 8.4.8 defItem

#### 8.4.8.1 Synopsis

```
AIT_Item defItem(string $l)
```

#### 8.4.8.2 Description

Retourne un item associé au type d'item courant. Si l'item n'existe pas il est automatiquement créé.

#### 8.4.8.3 Paramètres

string <b>\$l</b>	nom de l'item
-------------------	---------------

#### 8.4.8.4 Retour

Un objet AIT\_Item représentant l'item trouvé.

### 8.4.9 getItems

#### 8.4.9.1 Synopsis

```
AITResult.getItems(integer $offset=null, integer $lines=null, integer $ordering=null)
```

#### 8.4.9.2 Description

Retourne tous les items du type d'item courant

#### 8.4.9.3 Paramètres

integer <b>\$offset</b>	décalage à partir du premier enregistrement
integer <b>\$lines</b>	nombre de lignes à retourner
integer <b>\$ordering</b>	flag permettant le tri

#### 8.4.9.4 Retour

un objet AITResult (ArrayObject) contenant des objets AIT\_Item.

#### 8.4.9.5 Exemple

```
<?php
require_once 'prepend.php';

$items = $schema->disques->getItems();

foreach ($items as $item) {
    echo "- ".$item->get()."\n";
}
echo 'Showed : '.$items->count(). ' / '.$items->total()."\n";

// Affichera :
//
// - a2407474
// - a2274830
// - a1902425
// - a762424
// - a1095444
// (...)
```

### 8.4.10 fetchItems

#### 8.4.10.1 Synopsis

```
AITResult fetchItems(ArrayObject $tags, integer $offset=null, integer $lines=null, integer
```

#### 8.4.10.2 Description

Retourne les items possédant un ou plusieurs tags donnés en paramètres

#### 8.4.10.3 Paramètres

ArrayObject <b>\$tags</b>	Tableau de tag
integer <b>\$offset</b>	décalage à partir du premier enregistrement
integer <b>\$lines</b>	nombre de lignes à retourner
integer <b>\$ordering</b>	flag permettant le tri

#### 8.4.10.4 Retour

un objet AITResult (ArrayObject) contenant des objets AIT\_Item.

#### 8.4.10.5 Exemple

```
<?php
require_once 'prepend.php';

$jazz = $schema->style->getTag('Jazz');

$tags = new ArrayObject();
$tags->append($jazz);

$items = $schema->disques->fetchItems($tags, 0, 10);

foreach($items as $item) {
    echo $item->get()."\n";
}

echo 'Showed : ' . $items->count() . ' / ' . $items->total()."\n";
?>
```

### 8.4.11 getTagTypes

#### 8.4.11.1 Synopsis

```
AITResult getTagTypes(integer $offset=null, integer $lines=null, integer $ordering=null)
```

#### 8.4.11.2 Description

Retourne tous les types de tags de l'item courant

#### 8.4.11.3 Paramètres

integer <b>\$offset</b>	décalage à partir du premier enregistrement
integer <b>\$lines</b>	nombre de lignes à retourner
integer <b>\$ordering</b>	flag permettant le tri

#### 8.4.11.4 Retour

un objet AITResult (ArrayObject) contenant des objets AIT\_TagType.

### 8.4.12 searchItems

#### 8.4.12.1 Synopsis

```
AITResult searchItems(mixed $query, integer $offset=null, integer $lines=null, integer $or
```

#### 8.4.12.2 Description

Recherche des items du type courant

#### 8.4.12.3 Paramètres

string <b>\$query</b>	requete (le format peut varier en fonction de la callback) sans callback c'est une chaine de caractères contenant du SQL
integer <b>\$offset</b>	décalage à partir du premier enregistrement
integer <b>\$lines</b>	nombre de lignes à retourner
integer <b>\$ordering</b>	flag permettant le tri

#### 8.4.12.4 Retour

un objet AITResult (ArrayObject) contenant des objets AIT\_Item.

### 8.4.13 getItemBySystemID

#### 8.4.13.1 Synopsis

```
AIT_Item getItemBySystemID(integer $i)
```

#### 8.4.13.2 Description

Récupère un Item en fonction de son identifiant système

#### 8.4.13.3 Paramètres

integer <b>\$i</b>	un identifiant système
--------------------	------------------------

#### 8.4.13.4 Retour

Un objet AIT\_Item représentant l'item trouvé.

#### 8.4.13.5 Exemple

```
<?php
require_once 'prepend.php';

$itemA = $schema->disques->getItem('a1902425');
$id = $itemA->getSystemID();
```

```
$itemB = $schema->disques->getItemBySystemID($id);

if (!is_null($itemB)) {
    echo $itemB->get();
}
// Affichera : a1902425
```

### 8.4.14 queryItems

#### 8.4.14.1 Synopsis

```
AIT_ItemType::queryItems(AITQuery $query, $offset=null, $lines=null, $ordering=null)
```

#### 8.4.14.2 Description

On recherche des items associés au TYPE d'ITEM courant à partir d'un objet AITQuery

#### 8.4.14.3 Paramètres

<b>\$l</b>	\$query objet contenant la requête d'interrogation.
<b>\$offset</b>	décalage à partir du premier enregistrement.
<b>\$lines</b>	nombre de lignes à retourner.
<b>\$ordering</b>	flag permettant le tri.

#### 8.4.14.4 Retour

un objet AITResult (ArrayObject) contenant des objets AIT\_Item.

#### 8.4.14.5 Exemple

```
<?php
require_once 'prepend.php';

$jazz    = $schema->style->getTag('Jazz');
$blues   = $schema->style->getTag('Blues');
$selectro = $schema->style->getTag('Electro');

$query = new AITQuery($db);

// les disques marqués avec Jazz et Blues
$query->all(new ArrayObject(array($jazz, $blues)));

// Ou bien
$query->eitheror();
```

```
// Les disques marqués avec Jazz et Electro
$query->all(new ArrayObject(array($jazz, $electro)));

$items = $schema->disques->queryItems($query, 0, 10, AIT::ORDER_BY_SCORE|AIT::ORDER_ASC);
foreach($items as $item) {
    echo $item->get();
}
echo 'Showed : ' . $items->count() . ' / ' . $items->total() . "\n";
?>
```

## 8.4.15 countItems

### 8.4.15.1 Synopsis

```
AIT_ItemType::countItems()
```

### 8.4.15.2 Description

Retourne le nombre total d'items attachés au type d'item courant.

### 8.4.15.3 Paramètres

Aucun.

### 8.4.15.4 Retour

un nombre entier représentant le nombre d'enregistrement.

### 8.4.15.5 Exemple

```
<?php
require_once 'prepend.php';

$counter = $schema->disques->countItems();

echo $counter;

// Affiche le nombre de disque
```

## 8.4.16 countTagTypes

### 8.4.16.1 Synopsis

```
AIT_ItemType::countTags()
```

### 8.4.16.2 Description

Retourne le nombre total de tags attachés au type d'item courant.

### 8.4.16.3 Paramètres

Aucun.

#### **8.4.16.4 Retour**

un nombre entier représentant le nombre d'enregistrement.

### **8.4.17 del**

#### **8.4.17.1 Synopsis**

```
void del()
```

#### **8.4.17.2 Description**

Suppression de l'élément des tags et des items qui lui sont associés.

#### **8.4.17.3 Paramètres**

Aucun.

#### **8.4.17.4 Retour**

NULL.

### **8.4.18 getPDO**

#### **8.4.18.1 Synopsis**

```
PDO getPDO()
```

#### **8.4.18.2 Description**

Retourne l'objet PDOAIT utilisé dans l'objet

#### **8.4.18.3 Paramètres**

Aucun.

#### **8.4.18.4 Retour**

Un objet PDO

### **8.4.19 ren**

#### **8.4.19.1 Synopsis**

```
void ren(string $l)
```

#### **8.4.19.2 Description**

Renomme l'élément. Cela revient à changer son label.

#### **8.4.19.3 Paramètres**

---



string \$l

nouveau label

#### 8.4.19.4 Retour

NULL.

### 8.4.20 exists

#### 8.4.20.1 Synopsis

```
boolean exists()
```

#### 8.4.20.2 Description

Vérifie l'existence de l'élément

#### 8.4.20.3 Paramètres

Aucun.

#### 8.4.20.4 Retour

TRUE ou FALSE en fonction de l'existence de l'élément

### 8.4.21 getSystemID

#### 8.4.21.1 Synopsis

```
integer getSystemID()
```

#### 8.4.21.2 Description

Retourne l'identifiant système de l'élément.

#### 8.4.21.3 Paramètres

Aucun.

#### 8.4.21.4 Retour

Un nombre entier représentant l'identifiant système.

```
<?php
require_once 'prepend.php';

$item = $schema->disques->getItem('a1902425');
$id = $item->getSystemID();

var_dump($id);

// Affichera : int(17)
```

## 8.4.22 get

### 8.4.22.1 Synopsis

```
mixed get(string $name= 'label')
```

### 8.4.22.2 Description

Retourne la valeur d'une propriété de l'élément. Les propriétés accessibles sont : type, label, buffer, prefix, suffix

### 8.4.22.3 Paramètres

string <b>\$name</b>	nom de la propriété
----------------------	---------------------

### 8.4.22.4 Retour

mixed

## 8.4.23 set

### 8.4.23.1 Synopsis

```
mixed set(string $value, string $name= 'label')
```

### 8.4.23.2 Description

Fixe la valeur d'une propriété de l'élément. Les propriétés accessibles sont : type, label, buffer, prefix, suffix

### 8.4.23.3 Paramètres

string <b>\$value</b>	valeur de la propriété
-----------------------	------------------------

string <b>\$name</b>	nom de la propriété
----------------------	---------------------

### 8.4.23.4 Retour

Aucun.

## 8.4.24 search

### 8.4.24.1 Synopsis

```
AITResult search(mixed $query, integer $offset=null, integer $lines=null, integer $orderin
```

#### 8.4.24.2 Description

Recherche des éléments (quelque soit le type)

#### 8.4.24.3 Paramètres

string <b>\$query</b>	requete (le format peut varier en fonction de la callback) sans callback c'est une chaine de caractères contenant du SQL
integer <b>\$offset</b>	décalage à partir du premier enregistrement
integer <b>\$lines</b>	nombre de lignes à retourner
integer <b>\$ordering</b>	flag permettant le tri

#### 8.4.24.4 Retour

un objet AITResult (ArrayObject) contenant des objets AIT\_Tag ou AIT\_Item ou AIT\_TagType ou AIT\_ItemType.

### 8.4.25 getTimestamps

#### 8.4.25.1 Synopsis

```
ArrayObject getTimestamps()
```

#### 8.4.25.2 Description

Retourne la date de dernière modification et la date de création de l'élément. les différents "temps" enregistré pour l'élément

#### 8.4.25.3 Paramètres

Aucun.

#### 8.4.25.4 Retour

Un objet ArrayObject dont le premier item contient la date de modification et le deuxième contient la date de création.

#### 8.4.25.5 Exemple

```
<?php
require_once 'prepend.php';

$item = $schema->disques->getItem('a1902425');
$timestamps = $item->getTimestamps();
echo date('Y-m-d\TH:i:s\Z', $timestamps->updated)."\n";
// Affichera : 2008-08-21T17:22:17Z
echo $timestamps['created']."\n";
```

```
// Affichera : 1219332137

$jazz = $schema->style->getTag('Jazz');
$timestamps = $jazz->getTimestamps();
var_dump($timestamps);

// Affichera :
// object(ArrayObject)#15 (2) {
//     ["updated"]=>
//     int(1219332137)
//     ["created"]=>
//     int(1219332137)
// }
//
```

## 8.4.26 getScore

### 8.4.26.1 Synopsis

```
integer getScore()
```

### 8.4.26.2 Description

Retourne le score de l'élément

### 8.4.26.3 Paramètres

Aucun.

### 8.4.26.4 Retour

Un entier représentant le score l'élément.

## 8.4.27 setScore

### 8.4.27.1 Synopsis

```
void setScore($i)
```

### 8.4.27.2 Description

Donne un score à l'élément

### 8.4.27.3 Paramètres

integer <b>\$i</b>	Un entier représentant le score l'élément.
--------------------	--

#### 8.4.27.4 Retour

NULL.

### 8.4.28 setFrequency

#### 8.4.28.1 Synopsis

```
void setFrequency($i)
```

#### 8.4.28.2 Description

Donne une fréquence à l'élément

#### 8.4.28.3 Paramètres

integer <b>\$i</b>	Un entier représentant la fréquence de l'élément
--------------------	--

#### 8.4.28.4 Retour

NULL.

### 8.4.29 debug

#### 8.4.29.1 Synopsis

```
void debug()
```

#### 8.4.29.2 Description

DEBUG

#### 8.4.29.3 Paramètres

Aucun.

#### 8.4.29.4 Retour

NULL.

### 8.4.30 dump

#### 8.4.30.1 Synopsis

```
mixed dump(string $s = '', boolean $r = false)
```

---

#### 8.4.30.2 Description

Afficher le contenu de l'objet. Si \$r est TRUE. Alors le contenu de l'objet précédé éventuellement de \$s seront affichée entouré des <pre> et </pre>

#### 8.4.30.3 Paramètres

string \$s	chaîne de caractères à afficher avant le contenu de l'objet.
boolean \$r	si vrai retourne la chaîne à afficher plutôt que de l'afficher

#### 8.4.30.4 Retour

Retourne NULL ou une chaîne de caractères (dépend du paramètre \$r).

### 8.4.31 getBySystemID

#### 8.4.31.1 Synopsis

```
mixed AIT::getBySystemID(PDOAIT $pdo, $id)
```

#### 8.4.31.2 Description

Crée l'objet correspondant à un identifiant système

#### 8.4.31.3 Paramètres

PDOAIT \$pdo	pointeur sur la base de données
integer \$id	identifiant système

#### 8.4.31.4 Retour

Retourne NULL ou une instance d'objet (AIT\_ItemType, AIT\_Item, AIT\_TagType, AIT\_Tag)

#### 8.4.31.5 Exemple

```
<?php
require_once 'prepend.php';

$o = AIT::getBySystemID($db, 17);
echo $o->get() . " : ";

$o = AIT::getBySystemID($db, 18);
echo $o->get() . ", ";
```

```
$o = AIT::getBySystemID($db, 19);
echo $o->get() . "\n";

// Affichera : a1902425 : Heroes, Antonio Vivaldi
```

## 8.5 AIT\_Item

Cette classe permet de manipuler un ITEM.

### 8.5.1 \_\_construct

#### 8.5.1.1 Synopsis

```
void __construct(string $l, integer $t, PDOAIT $pdo, integer $id=false)
```

#### 8.5.1.2 Description

Constructeur

#### 8.5.1.3 Paramètres

string <b>\$l</b>	nom du nouveau item
integer <b>\$t</b>	Identifiant système du type d'item associé (si connu)
PDOAIT <b>\$pdo</b>	Instance de base AIT que l'on souhaite utiliser.
integer <b>\$id</b>	Identifiant système de l'élément (si déjà connu).

#### defTag

Synopsis  
+++++

```
AIT_Tag defTag(string $l, AIT_TagType $o)
```

Description  
+++++

Retourne un tag associé à l'item courant.  
Si le Tag n'existe pas il est automatiquement créé.

Paramètres  
+++++

```
string *$l*:: nom du tag
AIT_TagType *$o*:: Type de Tag
```

```
Retour  
+++++
```

Un objet AIT\_Tag représentant le tag.

```
// -----  
getTag
```

#### 8.5.1.4 Synopsis

```
AIT_Tag getTag(string $l, AIT_TagType $o)
```

#### 8.5.1.5 Description

Retourne un tag associé à l'item courant (si il existe)

#### 8.5.1.6 Paramètres

string <b>\$l</b>	nom du tag
AIT_TagType <b>\$o</b>	Type de Tag

#### 8.5.1.7 Retour

Un objet AIT\_Tag représentant le tag trouvé ou NULL si il n'existe pas.

### 8.5.2 addTag

#### 8.5.2.1 Synopsis

```
AIT_Tag addTag(string $l, AIT_TagType $o)
```

#### 8.5.2.2 Description

Ajout d'un tag à l'item courant

#### 8.5.2.3 Paramètres

string <b>\$l</b>	nom du nouveau tag
AIT_TagType <b>\$o</b>	Type de Tag



#### 8.5.2.4 Retour

Un objet AIT\_Tag représentant le nouveau tag.

### 8.5.3 delTag

#### 8.5.3.1 Synopsis

```
delTag(string $l, AIT_TagType $o)
```

#### 8.5.3.2 Description

Supprime un tag à l'item courant

#### 8.5.3.3 Paramètres

string <b>\$l</b>	nom du nouveau tag
AIT_TagType <b>\$o</b>	Type de Tag

#### 8.5.3.4 Retour

NULL.

### 8.5.4 attach

#### 8.5.4.1 Synopsis

```
AIT_Item attach(AIT_Tag $o, mixed $r = null)
```

#### 8.5.4.2 Description

Ajoute une association entre le tag et un item. On peut préciser l'ordre des tags en indiquant un tag après lequel on doit s'insérer ou la constante AIT::INSERT\_FIRST. Dans ce cas on ajoutera le tag en premier dans la liste des tags associés à AIT\_Item. Si \$r reste à NULL le tag sera ajouté en fin de liste.

#### 8.5.4.3 Paramètres

AIT_Tag <b>\$o</b>	Tag
mixed <b>\$r</b>	Un objet AIT_Tag ou AIT::INSERT_FIRST

#### 8.5.4.4 Retour

Retourne l'objet en lui-même. Ce qui permet d'enchaîner les méthodes.

### 8.5.5 detach

#### 8.5.5.1 Synopsis

```
AIT_Item detach(AIT_Tag $o)
```

#### 8.5.5.2 Description

Supprime une association entre le tag et un item

#### 8.5.5.3 Paramètres

AIT_Tag \$o	Tag
-------------	-----

#### 8.5.5.4 Retour

Retourne l'objet en lui-même. Ce qui permet d'enchaîner les méthodes.

### 8.5.6 getTags

#### 8.5.6.1 Synopsis

```
AITResult getTags(integer $offset=null, integer $lines=null, integer $ordering=null)
```

#### 8.5.6.2 Description

Retourne tous les tags de l'item courant

#### 8.5.6.3 Paramètres

integer \$offset	décalage à partir du premier enregistrement
integer \$lines	nombre de lignes à retourner
integer \$ordering	flag permettant le tri

#### 8.5.6.4 Retour

Un objet AITResult (ArrayObject) contenant des objets AIT\_Tag.

---

## 8.5.7 getTypedTags

### 8.5.7.1 Synopsis

```
AIT_Item::getTypedTags(AIT_TagType $typetag, $offset=null, $lines=null, $ordering=null)
```

### 8.5.7.2 Description

Retourne tous les tags d'un certain type de l'item courant

### 8.5.7.3 Paramètres

AIT_TagType <b>\$typetag</b>	un type de tag
integer <b>\$offset</b>	décalage à partir du premier enregistrement
integer <b>\$lines</b>	nombre de lignes à retourner
integer <b>\$ordering</b>	flag permettant le tri

### 8.5.7.4 Retour

Un objet AITResult (ArrayObject) contenant des objets AIT\_Tag.

### 8.5.7.5 Exemple

```
<?php
require_once 'prepend.php';

$item = $schema->disques->getItem('a1902425');

$tags = $item->getTypedTags($schema->style);

if ($tags->count() > 0) {
    echo "- ".$tags->offsetGet(0)->get()."\n";
}
echo 'Found : '.$tags->count(). ' / '.$tags->total()."\n";
```

## 8.5.8 fetchTags

### 8.5.8.1 Synopsis

```
AITResult fetchTags(ArrayObject $tags, integer $offset=null, integer $lines=null, integer
```

### 8.5.8.2 Description

Retourne les tags possédant de un ou plusieurs tags donnés

### 8.5.8.3 Paramètres

ArrayObject <b>\$tags</b>	Tableau de type de tag
integer <b>\$offset</b>	décalage à partir du premier enregistrement
integer <b>\$lines</b>	nombre de lignes à retourner
integer <b>\$ordering</b>	flag permettant le tri

### 8.5.8.4 Retour

Un objet AITResult (ArrayObject) contenant des objets AIT\_Tag.

## 8.5.9 getItems

### 8.5.9.1 Synopsis

```
AITResult getItems(integer $offset=null, integer $lines=null, integer $ordering=null)
```

### 8.5.9.2 Description

Retourne tous les items attachés à l'item courant

### 8.5.9.3 Paramètres

integer <b>\$offset</b>	décalage à partir du premier enregistrement
integer <b>\$lines</b>	nombre de lignes à retourner
integer <b>\$ordering</b>	flag permettant le tri

### 8.5.9.4 Retour

Un objet AITResult (ArrayObject) contenant des objets AIT\_Item.

## 8.5.10 getTypedItems

### 8.5.10.1 Synopsis

```
AIT_Item::getTypedItems(AIT_ItemType $typeitem, $offset=null, $lines=null, $ordering=null)
```

### 8.5.10.2 Description

Retourne tous les items d'un certain type attaché à l'item courant

### 8.5.10.3 Paramètres

AIT_ItemType <b>\$itemtag</b>	un type d'item
integer <b>\$offset</b>	décalage à partir du premier enregistrement
integer <b>\$lines</b>	nombre de lignes à retourner
integer <b>\$ordering</b>	flag permettant le tri

### 8.5.10.4 Retour

Un objet AITResult (ArrayObject) contenant des objets AIT\_Item.

## 8.5.11 fetchItems

### 8.5.11.1 Synopsis

```
AITResult fetchItems(ArrayObject $types, integer $offset=null, integer $lines=null, integer $ordering=null)
```

### 8.5.11.2 Description

Retourne les items attachés à l'item courant à condition qu'il soit de l'un des types d'item donné dans \$types

### 8.5.11.3 Paramètres

ArrayObject <b>\$tags</b>	Tableau de type d'item
integer <b>\$offset</b>	décalage à partir du premier enregistrement
integer <b>\$lines</b>	nombre de lignes à retourner
integer <b>\$ordering</b>	flag permettant le tri

### 8.5.11.4 Retour

Un objet AITResult (ArrayObject) contenant des objets AIT\_Item.  
getElements

**Synopsis**

++++++

```
AITResult getElements(integer $offset=null, integer $lines=null, integer $ordering=null ←
)
```

**Description**

++++++

Retourne tous les éléments attachés à l'*item* courant

**Paramètres**

++++++

```
integer *$offset*:: décalage à partir du premier enregistrement
integer *$lines*:: nombre de lignes à retourner
integer *$ordering*:: flag permettant le tri
```

**Retour**

+++++

Un objet AITResult (ArrayObject) contenant des objets AIT\_Item et/ou AIT\_Tag.

```
// -----
getTypedElements
```

**8.5.11.5 Synopsis**

```
AIT_Item::getTypedElements($type, $offset=null, $lines=null, $ordering=null)
```

**8.5.11.6 Description**

Retourne tous les éléments d'un certain type attaché à l'*item* courant

**8.5.11.7 Paramètres**

mixed <b>\$type</b>	un type d'item ou un type de tag
integer <b>\$offset</b>	décalage à partir du premier enregistrement
integer <b>\$lines</b>	nombre de lignes à retourner
integer <b>\$ordering</b>	flag permettant le tri

**8.5.11.8 Retour**

Un objet AITResult (ArrayObject) contenant des objets AIT\_Item et/ou AIT\_Tag.

## 8.5.12 fetchItems

### 8.5.12.1 Synopsis

```
AITResult fetchElements(ArrayObject $types, integer $offset=null, integer $lines=null, int
```

### 8.5.12.2 Description

Retourne les éléments attachés à l'item courant à condition qu'il soit de l'un des types d'item donné dans \$types

### 8.5.12.3 Paramètres

ArrayObject <b>\$tags</b>	Tableau de type d'item
integer <b>\$offset</b>	décalage à partir du premier enregistrement
integer <b>\$lines</b>	nombre de lignes à retourner
integer <b>\$ordering</b>	flag permettant le tri
integer <b>\$control</b>	permet de filtrer suivant les tags (AIT::TAG) ou les items (AIT::ITEM)

### 8.5.12.4 Retour

Un objet AITResult (ArrayObject) contenant des objets AIT\_Item et/ou AIT\_Tag.

## 8.5.13 del

### 8.5.13.1 Synopsis

```
void del(boolean $cascade = false)
```

### 8.5.13.2 Description

Suppression de l'élément. On peut également avec le paramètre \$cascade supprimer les tags associés.

### 8.5.13.3 Paramètres

boolean <b>\$cascade</b>	permet une suppression en cascade
--------------------------	-----------------------------------

### 8.5.13.4 Retour

NULL.

---

### 8.5.14 getItemType

#### 8.5.14.1 Synopsis

```
AIT_ItemType getItemType ()
```

#### 8.5.14.2 Description

Retourne le type d'item associé à l'élément courant

#### 8.5.14.3 Paramètres

Aucun.

#### 8.5.14.4 Retour

Un objet AIT\_ItemType représentant le type d'item associé.

### 8.5.15 getTagsObject

#### 8.5.15.1 Synopsis

```
AITTagsObject getTagsObject ()
```

#### 8.5.15.2 Description

Retourne tous les tags de l'élément courant sous forme d'un objet.

#### 8.5.15.3 Paramètres

Aucun.

#### 8.5.15.4 Retour

Un objet AITTagsObject représentant un ensemble de tag trié par type

#### 8.5.15.5 Exemple

```
<?php
require_once 'prepend.php';

$item = $schema->disques->getItem('a1902425');

$tct = $item->getTagsObject();

echo $tct->titre->offsetGet(0)->get() . "\n";
# Affichera : Heroes

foreach($tct->artiste as $tag)
    echo ' * ' . $tag->get();

# Affichera : * Antonio Vivaldi * Philippe Jaroussky
```



## 8.5.16 getPDO

### 8.5.16.1 Synopsis

```
PDO getPDO()
```

### 8.5.16.2 Description

Retourne l'objet PDOAIT utilisé dans l'objet

### 8.5.16.3 Paramètres

Aucun.

### 8.5.16.4 Retour

Un objet PDO

## 8.5.17 ren

### 8.5.17.1 Synopsis

```
void ren(string $l)
```

### 8.5.17.2 Description

Renomme l'élément. Cela revient à changer son label.

### 8.5.17.3 Paramètres

string \$l	nouveau label
------------	---------------

### 8.5.17.4 Retour

NULL.

## 8.5.18 exists

### 8.5.18.1 Synopsis

```
boolean exists()
```

### 8.5.18.2 Description

Vérifie l'existence de l'élément

---

### 8.5.18.3 Paramètres

Aucun.

### 8.5.18.4 Retour

TRUE ou FALSE en fonction de l'existence de l'élément

## 8.5.19 getSystemID

### 8.5.19.1 Synopsis

```
integer getSystemID()
```

### 8.5.19.2 Description

Retourne l'identifiant système de l'élément.

### 8.5.19.3 Paramètres

Aucun.

### 8.5.19.4 Retour

Un nombre entier représentant l'identifiant système.

```
<?php
require_once 'prepend.php';

$item = $schema->disques->getItem('a1902425');
$id = $item->getSystemID();

var_dump($id);

// Affichera : int(17)
```

## 8.5.20 get

### 8.5.20.1 Synopsis

```
mixed get(string $name= 'label')
```

### 8.5.20.2 Description

Retourne la valeur d'une propriété de l'élément. Les propriétés accessibles sont : type, label, buffer, prefix, suffix

### 8.5.20.3 Paramètres

string <b>\$name</b>	nom de la propriété
----------------------	---------------------

#### 8.5.20.4 Retour

mixed

### 8.5.21 set

#### 8.5.21.1 Synopsis

```
mixed set(string $value, string $name= 'label')
```

#### 8.5.21.2 Description

Fixe la valeur d'une propriété de l'élément. Les propriétés accessibles sont : type, label, buffer, prefix, suffix

#### 8.5.21.3 Paramètres

string <b>\$value</b>	valeur de la propriété
string <b>\$name</b>	nom de la propriété

#### 8.5.21.4 Retour

Aucun.

### 8.5.22 search

#### 8.5.22.1 Synopsis

```
AITResult search(mixed $query, integer $offset=null, integer $lines=null, integer $ordering)
```

#### 8.5.22.2 Description

Recherche des éléments (quelque soit le type)

#### 8.5.22.3 Paramètres

string <b>\$query</b>	requete (le format peut varier en fonction de la callback) sans callback c'est une chaîne de caractères contenant du SQL
integer <b>\$offset</b>	décalage à partir du premier enregistrement
integer <b>\$lines</b>	nombre de lignes à retourner
integer <b>\$ordering</b>	flag permettant le tri

#### 8.5.22.4 Retour

un objet AITResult (ArrayObject) contenant des objets AIT\_Tag ou AIT\_Item ou AIT\_TagType ou AIT\_ItemType.

### 8.5.23 getTimestamps

#### 8.5.23.1 Synopsis

```
ArrayObject getTimestamps()
```

#### 8.5.23.2 Description

Retourne la date de dernière modification et la date de création de l'élément. les différents "temps" enregistré pour l'élément

#### 8.5.23.3 Paramètres

Aucun.

#### 8.5.23.4 Retour

Un objet ArrayObject dont le premier item contient la date de modification et le deuxième contient la date de création.

#### 8.5.23.5 Exemple

```
<?php
require_once 'prepend.php';

$item = $schema->disques->getItem('a1902425');
$timestamps = $item->getTimestamps();
echo date('Y-m-d\TH:i:s\Z', $timestamps->updated)."\n";
// Affichera : 2008-08-21T17:22:17Z
echo $timestamps['created']."\n";
// Affichera : 1219332137

$jazz = $schema->style->getTag('Jazz');
$timestamps = $jazz->getTimestamps();
var_dump($timestamps);

// Affichera :
// object(ArrayObject)#15 (2) {
//     ["updated"]=>
//     int(1219332137)
//     ["created"]=>
//     int(1219332137)
// }
```

### 8.5.24 getScore

#### 8.5.24.1 Synopsis

```
integer getScore()
```

#### 8.5.24.2 Description

Retourne le score de l'élément

#### 8.5.24.3 Paramètres

Aucun.

#### 8.5.24.4 Retour

Un entier représentant le score l'élément.

### 8.5.25 setScore

#### 8.5.25.1 Synopsis

```
void setScore($i)
```

#### 8.5.25.2 Description

Donne un score à l'élément

#### 8.5.25.3 Paramètres

integer <b>\$i</b>	Un entier représentant le score l'élément.
--------------------	--

#### 8.5.25.4 Retour

NULL.

### 8.5.26 setFrequency

#### 8.5.26.1 Synopsis

```
void setFrequency($i)
```

#### 8.5.26.2 Description

Donne une fréquence à l'élément

#### 8.5.26.3 Paramètres

integer <b>\$i</b>	Un entier représentant la fréquence de l'élément
--------------------	--

---

#### 8.5.26.4 Retour

NULL.

### 8.5.27 debug

#### 8.5.27.1 Synopsis

```
void debug()
```

#### 8.5.27.2 Description

DEBUG

#### 8.5.27.3 Paramètres

Aucun.

#### 8.5.27.4 Retour

NULL.

### 8.5.28 dump

#### 8.5.28.1 Synopsis

```
mixed dump(string $s = '', boolean $r = false)
```

#### 8.5.28.2 Description

Afficher le contenu de l'objet. Si \$r est TRUE. Alors le contenu de l'objet précédé éventuellement de \$s seront affichée entouré des <pre> et </pre>

#### 8.5.28.3 Paramètres

string \$s	chaîne de caractères à afficher avant le contenu de l'objet.
boolean \$r	si vrai retourne la chaîne à afficher plutôt que de l'afficher

#### 8.5.28.4 Retour

Retourne NULL ou une chaîne de caractères (dépend du paramètre \$r).

## 8.5.29 getBySystemID

### 8.5.29.1 Synopsis

```
mixed AIT::getBySystemID(PDOAIT $pdo, $id)
```

### 8.5.29.2 Description

Cr  e l'objet correspondant   un identifiant syst  me

### 8.5.29.3 Param  tres

PDOAIT **\$pdo**                      pointeur sur la base de donn  es

integer **\$id**                      identifiant syst  me

### 8.5.29.4 Retour

Retourne NULL ou une instance d'objet (AIT\_ItemType, AIT\_Item, AIT\_TagType, AIT\_Tag)

### 8.5.29.5 Exemple

```
<?php
require_once 'prepend.php';

$o = AIT::getBySystemID($db, 17);
echo $o->get() . " : ";

$o = AIT::getBySystemID($db, 18);
echo $o->get() . ", ";

$o = AIT::getBySystemID($db, 19);
echo $o->get() . "\n";

// Affichera : a1902425 : Heroes, Antonio Vivaldi
```

## 8.6 AIT\_TagType

Cette classe permet de manipuler un "type de tag".

### 8.6.1 \_\_construct

#### 8.6.1.1 Synopsis

```
void __construct(string $l, integer $i, AITPDO $pdo, integer $id = false)
```

#### 8.6.1.2 Description

Constructeur

### 8.6.1.3 Paramètres

string <b>\$l</b>	Label du type de tag.
integer <b>\$i</b>	Identifiant système du type d'item associé.
AITPDO <b>\$pdo</b>	L'instance de base AIT que l'on souhaite utiliser.
integer <b>\$id</b>	Identifiant système de l'élément (si déjà connu) .

## 8.6.2 newTag

### 8.6.2.1 Synopsis

```
AIT_Tag newTag()
```

### 8.6.2.2 Description

Crée un tag. Son label est calculé automatiquement en fonction de l'identifiant système.

### 8.6.2.3 Paramètres

Aucun.

### 8.6.2.4 Retour

un objet AIT\_Tag.

### 8.6.2.5 Exemple

```
<?php
require_once 'prepend.php';

// Ajout d'un nouveau tag
$tag = $schema->style->newTag();

echo $tag->get();
```

### defTag

Synopsis  
+++++

```
AIT_Tag defTag(string $l)
```

Description  
+++++



Récupère un tag du type courant. Si le tag n'existe pas, il est automatiquement créé.

Paramètres

+++++

string \*\$l\*:: nom du nouveau tag

Retour

+++++

Un objet AIT\_Tag représentant le tag

// -----  
addTag

### 8.6.2.6 Synopsis

```
AIT_Tag addTag(string $l)
```

### 8.6.2.7 Description

Ajout d'un tag au type de tag courant

### 8.6.2.8 Paramètres

string \$l	nom du nouveau tag
------------	--------------------

### 8.6.2.9 Retour

Un objet AIT\_Tag représentant le nouveau tag

## 8.6.3 getTag

### 8.6.3.1 Synopsis

```
AIT_Tag getTag(string $l)
```

### 8.6.3.2 Description

Récupère un tag

### 8.6.3.3 Paramètres

string \$l	nom du tag
------------	------------

#### 8.6.3.4 Retour

Un objet `AIT_Tag` représentant le tag trouvé ou `NULL` si il n'existe pas.

### 8.6.4 `getTags`

#### 8.6.4.1 Synopsis

```
AITResult getTags(integer $offset=null, integer $lines=null, integer $ordering=null)
```

#### 8.6.4.2 Description

Retourne tous les tags du type de tag courant.

#### 8.6.4.3 Paramètres

integer <code>\$offset</code>	décalage à partir du premier enregistrement.
integer <code>\$lines</code>	nombre de lignes à retourner.
integer <code>\$ordering</code>	flag permettant le tri.

#### 8.6.4.4 Retour

un objet `AITResult` (`ArrayObject`) contenant des objets `AIT_Tag`.

### 8.6.5 `getTagBySystemID`

#### 8.6.5.1 Synopsis

```
AIT_Tag getTagBySystemID(integer $i)
```

#### 8.6.5.2 Description

Récupère un tag avec son identifiant système.

#### 8.6.5.3 Paramètres

integer <code>\$i</code>	l'identifiant système.
--------------------------	------------------------

#### 8.6.5.4 Retour

Un objet `AIT_Tag` représentant le tag trouvé.

---

## 8.6.6 countItems

### 8.6.6.1 Synopsis

```
AIT_TagType::countItems()
```

### 8.6.6.2 Description

Retourne le nombre total de tags attachés au type de tag courant.

### 8.6.6.3 Paramètres

Aucun.

### 8.6.6.4 Retour

un nombre entier représentant le nombre d'enregistrement.

```
<?php
require_once 'prepend.php';

echo $schema->style->countItems();
```

## 8.6.7 del

### 8.6.7.1 Synopsis

```
void del(boolean $cascade = false)
```

### 8.6.7.2 Description

Suppression de l'élément et des tags qui lui sont associés. On peut également avec le paramètre \$cascade supprimer les items associés aux tags.

### 8.6.7.3 Paramètres

boolean <b>\$cascade</b>	permet une suppression en cascade
--------------------------	-----------------------------------

### 8.6.7.4 Retour

NULL.

## 8.6.8 searchTags

### 8.6.8.1 Synopsis

```
AITResult searchTags(mixed $query, integer $offset=null, integer $lines=null, integer $ord
```

### 8.6.8.2 Description

Recherche des tags du type courant

### 8.6.8.3 Paramètres

string <b>\$query</b>	requete (le format peut varier en fonction de la callback) sans callback c'est une chaîne de caractères contenant du SQL
integer <b>\$offset</b>	décalage à partir du premier enregistrement
integer <b>\$lines</b>	nombre de lignes à retourner
integer <b>\$ordering</b>	flag permettant le tri

### 8.6.8.4 Retour

un objet AITResult (ArrayObject) contenant des objets AIT\_Tag.

## 8.6.9 getPDO

### 8.6.9.1 Synopsis

```
PDO getPDO()
```

### 8.6.9.2 Description

Retourne l'objet PDOAIT utilisé dans l'objet

### 8.6.9.3 Paramètres

Aucun.

### 8.6.9.4 Retour

Un objet PDO

## 8.6.10 ren

### 8.6.10.1 Synopsis

```
void ren(string $l)
```

### 8.6.10.2 Description

Renomme l'élément. Cela revient à changer son label.

### 8.6.10.3 Paramètres

string \$l                                      nouveau label

### 8.6.10.4 Retour

NULL.

## 8.6.11 exists

### 8.6.11.1 Synopsis

```
boolean exists()
```

### 8.6.11.2 Description

Vérifie l'existence de l'élément

### 8.6.11.3 Paramètres

Aucun.

### 8.6.11.4 Retour

TRUE ou FALSE en fonction de l'existence de l'élément

## 8.6.12 getSystemID

### 8.6.12.1 Synopsis

```
integer getSystemID()
```

### 8.6.12.2 Description

Retourne l'identifiant système de l'élément.

### 8.6.12.3 Paramètres

Aucun.

### 8.6.12.4 Retour

Un nombre entier représentant l'identifiant système.

---

```
<?php
require_once 'prepend.php';

$item = $schema->disques->getItem('a1902425');
$id = $item->getSystemID();

var_dump($id);

// Affichera : int(17)
```

### 8.6.13 get

#### 8.6.13.1 Synopsis

```
mixed get(string $name= 'label')
```

#### 8.6.13.2 Description

Retourne la valeur d'une propriété de l'élément. Les propriétés accessibles sont : type, label, buffer, prefix, suffix

#### 8.6.13.3 Paramètres

string <b>\$name</b>	nom de la propriété
----------------------	---------------------

#### 8.6.13.4 Retour

mixed

### 8.6.14 set

#### 8.6.14.1 Synopsis

```
mixed set(string $value, string $name= 'label')
```

#### 8.6.14.2 Description

Fixe la valeur d'une propriété de l'élément. Les propriétés accessibles sont : type, label, buffer, prefix, suffix

#### 8.6.14.3 Paramètres

string <b>\$value</b>	valeur de la propriété
-----------------------	------------------------

string <b>\$name</b>	nom de la propriété
----------------------	---------------------

#### 8.6.14.4 Retour

Aucun.

### 8.6.15 search

#### 8.6.15.1 Synopsis

```
AITResult search(mixed $query, integer $offset=null, integer $lines=null, integer $ordering)
```

#### 8.6.15.2 Description

Recherche des éléments (quelque soit le type)

#### 8.6.15.3 Paramètres

string <b>\$query</b>	requete (le format peut varier en fonction de la callback) sans callback c'est une chaine de caractères contenant du SQL
integer <b>\$offset</b>	décalage à partir du premier enregistrement
integer <b>\$lines</b>	nombre de lignes à retourner
integer <b>\$ordering</b>	flag permettant le tri

#### 8.6.15.4 Retour

un objet AITResult (ArrayObject) contenant des objets AIT\_Tag ou AIT\_Item ou AIT\_TagType ou AIT\_ItemType.

### 8.6.16 getTimestamps

#### 8.6.16.1 Synopsis

```
ArrayObject getTimestamps()
```

#### 8.6.16.2 Description

Retourne la date de dernière modification et la date de création de l'élément. les différents "temps" enregistré pour l'élément

#### 8.6.16.3 Paramètres

Aucun.

#### 8.6.16.4 Retour

Un objet ArrayObject dont le premier item contient la date de modification et le deuxième contient la date de création.

---

### 8.6.16.5 Exemple

```
<?php
require_once 'prepend.php';

$item = $schema->disques->getItem('a1902425');
$timestamps = $item->getTimestamps();
echo date('Y-m-d\TH:i:s\Z', $timestamps->updated)."\n";
// Affichera : 2008-08-21T17:22:17Z
echo $timestamps['created']."\n";
// Affichera : 1219332137

$jazz = $schema->style->getTag('Jazz');
$timestamps = $jazz->getTimestamps();
var_dump($timestamps);

// Affichera :
// object(ArrayObject)#15 (2) {
//     ["updated"]=>
//     int(1219332137)
//     ["created"]=>
//     int(1219332137)
// }
```

## 8.6.17 getScore

### 8.6.17.1 Synopsis

```
integer getScore()
```

### 8.6.17.2 Description

Retourne le score de l'élément

### 8.6.17.3 Paramètres

Aucun.

### 8.6.17.4 Retour

Un entier représentant le score l'élément.

## 8.6.18 setScore

### 8.6.18.1 Synopsis

```
void setScore($i)
```

### 8.6.18.2 Description

Donne un score à l'élément



### **8.6.18.3 Paramètres**

integer \$i

Un entier représentant le score l'élément.

#### 8.6.18.4 Retour

NULL.

### 8.6.19 setFrequency

#### 8.6.19.1 Synopsis

```
void setFrequency($i)
```

#### 8.6.19.2 Description

Donne une fréquence à l'élément

#### 8.6.19.3 Paramètres

integer \$i

Un entier représentant la fréquence de l'élément

#### 8.6.19.4 Retour

NULL.

### 8.6.20 debug

#### 8.6.20.1 Synopsis

```
void debug()
```

#### 8.6.20.2 Description

DEBUG

#### 8.6.20.3 Paramètres

Aucun.

#### 8.6.20.4 Retour

NULL.

### 8.6.21 dump

#### 8.6.21.1 Synopsis

```
mixed dump(string $s = '', boolean $r = false)
```

---

### 8.6.21.2 Description

Afficher le contenu de l'objet. Si \$r est TRUE. Alors le contenu de l'objet précédé éventuellement de \$s seront affichée entouré des <pre> et </pre>

### 8.6.21.3 Paramètres

string \$s	chaîne de caractères à afficher avant le contenu de l'objet.
boolean \$r	si vrai retourne la chaîne à afficher plutôt que de l'afficher

### 8.6.21.4 Retour

Retourne NULL ou une chaîne de caractères (dépend du paramètre \$r).

## 8.6.22 getBySystemID

### 8.6.22.1 Synopsis

```
mixed AIT::getBySystemID(PDOAIT $pdo, $id)
```

### 8.6.22.2 Description

Crée l'objet correspondant à un identifiant système

### 8.6.22.3 Paramètres

PDOAIT \$pdo	pointeur sur la base de données
integer \$id	identifiant système

### 8.6.22.4 Retour

Retourne NULL ou une instance d'objet (AIT\_ItemType, AIT\_Item, AIT\_TagType, AIT\_Tag)

### 8.6.22.5 Exemple

```
<?php
require_once 'prepend.php';

$o = AIT::getBySystemID($db, 17);
echo $o->get() . " : ";

$o = AIT::getBySystemID($db, 18);
echo $o->get() . ", ";
```

```
$o = AIT::getBySystemID($db, 19);  
echo $o->get() . "\n";  
  
// Affichera : a1902425 : Heroes, Antonio Vivaldi
```

## 8.7 AIT\_Tag

Cette classe permet de manipuler un TAG.

### 8.7.1 \_\_construct

#### 8.7.1.1 Synopsis

```
void __construct($l, $t, $i, PDOAIT $pdo, $id=false)
```

#### 8.7.1.2 Description

Le constructeur de la classe.

#### 8.7.1.3 Paramètres

string <b>\$l</b>	Label du TAG
integer <b>\$t</b>	Identifiant système du TYPE de TAG associé.
integer <b>\$i</b>	Identifiant système de l'ITEM (si connu).
PDOAIT <b>\$pdo</b>	Instance de base AIT que l'on souhaite utiliser.
integer <b>\$id</b>	Identifiant système de l'élément (si déjà connu).

### 8.7.2 detach

#### 8.7.2.1 Synopsis

```
AIT_Tag detach()
```

#### 8.7.2.2 Description

Supprime l'association entre le tag son item

#### 8.7.2.3 Paramètres

Aucun.

#### 8.7.2.4 Retour

Retourne l'objet en lui-même. Ce qui permet d'enchaîner les méthodes.

#### 8.7.2.5 Exemple

```
<?php
require_once 'prepend.php';

$item = $schema->disques->getItem('a1902425');

$tags = $item->getTypedTags($schema->style);

foreach($tags as $tag) {
    $tag->detach();
}
// Supprime tous les tags "style" asscoié à a1902425
```

### 8.7.3 attach

#### 8.7.3.1 Synopsis

```
AIT_Tag attach(AIT_Item $o, Mixed $r = null)
```

#### 8.7.3.2 Description

Ajoute une association entre le tag et un item. On peut préciser l'ordre des tags en indiquant un tag après lequel on doit s'insérer ou la constante AIT::INSERT\_FIRST. Dans ce cas on ajoutera le tag en premier dans la liste des tags associés à AIT\_Item. Si \$r reste à NULL le tag sera ajouté en fin de liste.

#### 8.7.3.3 Paramètres

AIT_Item \$o	Un objet contenant un item.
mixed \$r	Un objet AIT_Tag ou AIT::INSERT_FIRST

#### 8.7.3.4 Retour

Retourne l'objet en lui-même. Ce qui permet d'enchaîner les méthodes.

```
<?php
require_once 'prepend.php';

$item = $schema->disques->getItem('a1902425');

$tag = $schema->style->addTag('disco');

// Ajout du tag "disco" à "a1902425"
$tag->attach($item);
```

## 8.7.4 getRelatedTags

### 8.7.4.1 Synopsis

```
AITResult getRelatedTags(integer $offset=null, integer $lines=null, integer $ordering=null)
```

### 8.7.4.2 Description

Retourne tous les tags associés au même item que le tag courant.

### 8.7.4.3 Paramètres

integer <b>\$offset</b>	décalage à partir du premier enregistrement
integer <b>\$lines</b>	nombre de lignes à retourner
integer <b>\$ordering</b>	flag permettant le tri.

### 8.7.4.4 Retour

Un objet AITResult (ArrayObject) contenant des objets AIT\_Tag.

## 8.7.5 fetchRelatedTags

### 8.7.5.1 Synopsis

```
AITResult fetchRelatedTags(ArrayObject $tags, integer $offset=null, integer $lines=null, integer $ordering=null)
```

### 8.7.5.2 Description

Retourne les tags associés au même item que le tag courant mais en filtrant sur un certain nombre de type de tag.

### 8.7.5.3 Paramètres

ArrayObject <b>\$tags</b>	Tableau de type de tag
integer <b>\$offset</b>	décalage à partir du premier enregistrement
integer <b>\$lines</b>	nombre de lignes à retourner
integer <b>\$ordering</b>	flag permettant le tri

#### 8.7.5.4 Retour

Un objet AITResult (ArrayObject) contenant des objets AIT\_Tag.

### 8.7.6 countItems

#### 8.7.6.1 Synopsis

```
integer countItems()
```

#### 8.7.6.2 Description

Retourne le nombre d'items attachés au tag courant.

#### 8.7.6.3 Paramètres

Aucun.

#### 8.7.6.4 Retour

Un entier représentant le nombre d'items

#### 8.7.6.5 Exemple

```
<?php
require_once 'prepend.php';

$jazz = $schema->style->getTag('Jazz');

if (!is_null($jazz))
    echo $jazz->countItems();

// Affichera : 2
```

### 8.7.7 del

#### 8.7.7.1 Synopsis

```
void del(boolean $cascade = false)
```

#### 8.7.7.2 Description

Suppression de l'élément. On peut également avec le paramètre \$cascade supprimer les items associés.

#### 8.7.7.3 Paramètres

boolean <b>\$cascade</b>	permet une suppression en cascade
--------------------------	-----------------------------------

#### 8.7.7.4 Retour

NULL.

### 8.7.8 getTagType

#### 8.7.8.1 Synopsis

```
AIT_TagType getTagType()
```

#### 8.7.8.2 Description

Retourne le type de tag associé à l'élément courant

#### 8.7.8.3 Paramètres

Aucun.

#### 8.7.8.4 Retour

Un objet AIT\_TagType représentant le type de tag associé.

### 8.7.9 getItems

#### 8.7.9.1 Synopsis

```
AITResult.getItems(integer $offset=null, integer $lines=null, integer $ordering=null)
```

#### 8.7.9.2 Description

Retourne tous les items du tag courant

#### 8.7.9.3 Paramètres

integer <b>\$offset</b>	décalage à partir du premier enregistrement
integer <b>\$lines</b>	nombre de lignes à retourner
integer <b>\$ordering</b>	flag permettant le tri

#### 8.7.9.4 Retour

un objet AITResult (ArrayObject) contenant des objets AIT\_Item.

---



## 8.7.10 getPDO

### 8.7.10.1 Synopsis

```
PDO getPDO()
```

### 8.7.10.2 Description

Retourne l'objet PDOAIT utilisé dans l'objet

### 8.7.10.3 Paramètres

Aucun.

### 8.7.10.4 Retour

Un objet PDO

## 8.7.11 ren

### 8.7.11.1 Synopsis

```
void ren(string $l)
```

### 8.7.11.2 Description

Renomme l'élément. Cela revient à changer son label.

### 8.7.11.3 Paramètres

string \$l	nouveau label
------------	---------------

### 8.7.11.4 Retour

NULL.

## 8.7.12 exists

### 8.7.12.1 Synopsis

```
boolean exists()
```

### 8.7.12.2 Description

Vérifie l'existence de l'élément

### 8.7.12.3 Paramètres

Aucun.

### 8.7.12.4 Retour

TRUE ou FALSE en fonction de l'existence de l'élément

## 8.7.13 getSystemID

### 8.7.13.1 Synopsis

```
integer getSystemID()
```

### 8.7.13.2 Description

Retourne l'identifiant système de l'élément.

### 8.7.13.3 Paramètres

Aucun.

### 8.7.13.4 Retour

Un nombre entier représentant l'identifiant système.

```
<?php
require_once 'prepend.php';

$item = $schema->disques->getItem('a1902425');
$id = $item->getSystemID();

var_dump($id);

// Affichera : int(17)
```

## 8.7.14 get

### 8.7.14.1 Synopsis

```
mixed get(string $name= 'label')
```

### 8.7.14.2 Description

Retourne la valeur d'une propriété de l'élément. Les propriétés accessibles sont : type, label, buffer, prefix, suffix

### 8.7.14.3 Paramètres

string <b>\$name</b>	nom de la propriété
----------------------	---------------------

#### 8.7.14.4 Retour

mixed

### 8.7.15 set

#### 8.7.15.1 Synopsis

```
mixed set(string $value, string $name= 'label')
```

#### 8.7.15.2 Description

Fixe la valeur d'une propriété de l'élément. Les propriétés accessibles sont : type, label, buffer, prefix, suffix

#### 8.7.15.3 Paramètres

string <b>\$value</b>	valeur de la propriété
string <b>\$name</b>	nom de la propriété

#### 8.7.15.4 Retour

Aucun.

### 8.7.16 search

#### 8.7.16.1 Synopsis

```
AITResult search(mixed $query, integer $offset=null, integer $lines=null, integer $ordering)
```

#### 8.7.16.2 Description

Recherche des éléments (quelque soit le type)

#### 8.7.16.3 Paramètres

string <b>\$query</b>	requete (le format peut varier en fonction de la callback) sans callback c'est une chaîne de caractères contenant du SQL
integer <b>\$offset</b>	décalage à partir du premier enregistrement
integer <b>\$lines</b>	nombre de lignes à retourner
integer <b>\$ordering</b>	flag permettant le tri

#### 8.7.16.4 Retour

un objet AITResult (ArrayObject) contenant des objets AIT\_Tag ou AIT\_Item ou AIT\_TagType ou AIT\_ItemType.

### 8.7.17 getTimestamps

#### 8.7.17.1 Synopsis

```
ArrayObject getTimestamps()
```

#### 8.7.17.2 Description

Retourne la date de dernière modification et la date de création de l'élément. les différents "temps" enregistré pour l'élément

#### 8.7.17.3 Paramètres

Aucun.

#### 8.7.17.4 Retour

Un objet ArrayObject dont le premier item contient la date de modification et le deuxième contient la date de création.

#### 8.7.17.5 Exemple

```
<?php
require_once 'prepend.php';

$item = $schema->disques->getItem('a1902425');
$timestamps = $item->getTimestamps();
echo date('Y-m-d\TH:i:s\Z', $timestamps->updated)."\n";
// Affichera : 2008-08-21T17:22:17Z
echo $timestamps['created']."\n";
// Affichera : 1219332137

$jazz = $schema->style->getTag('Jazz');
$timestamps = $jazz->getTimestamps();
var_dump($timestamps);

// Affichera :
// object(ArrayObject)#15 (2) {
//     ["updated"]=>
//     int(1219332137)
//     ["created"]=>
//     int(1219332137)
// }
```

### 8.7.18 getScore

#### 8.7.18.1 Synopsis

```
integer getScore()
```

#### 8.7.18.2 Description

Retourne le score de l'élément

#### 8.7.18.3 Paramètres

Aucun.

#### 8.7.18.4 Retour

Un entier représentant le score l'élément.

### 8.7.19 setScore

#### 8.7.19.1 Synopsis

```
void setScore($i)
```

#### 8.7.19.2 Description

Donne un score à l'élément

#### 8.7.19.3 Paramètres

integer <b>\$i</b>	Un entier représentant le score l'élément.
--------------------	--

#### 8.7.19.4 Retour

NULL.

### 8.7.20 setFrequency

#### 8.7.20.1 Synopsis

```
void setFrequency($i)
```

#### 8.7.20.2 Description

Donne une fréquence à l'élément

#### 8.7.20.3 Paramètres

integer <b>\$i</b>	Un entier représentant la fréquence de l'élément
--------------------	--

#### 8.7.20.4 Retour

NULL.

### 8.7.21 debug

#### 8.7.21.1 Synopsis

```
void debug()
```

#### 8.7.21.2 Description

DEBUG

#### 8.7.21.3 Paramètres

Aucun.

#### 8.7.21.4 Retour

NULL.

### 8.7.22 dump

#### 8.7.22.1 Synopsis

```
mixed dump(string $s = '', boolean $r = false)
```

#### 8.7.22.2 Description

Afficher le contenu de l'objet. Si \$r est TRUE. Alors le contenu de l'objet précédé éventuellement de \$s seront affichée entouré des <pre> et </pre>

#### 8.7.22.3 Paramètres

string <b>\$s</b>	chaîne de caractères à afficher avant le contenu de l'objet.
boolean <b>\$r</b>	si vrai retourne la chaîne à afficher plutôt que de l'afficher

#### 8.7.22.4 Retour

Retourne NULL ou une chaîne de caractères (dépend du paramètre \$r).

### 8.7.23 getBySystemID

#### 8.7.23.1 Synopsis

```
mixed AIT::getBySystemID(PDOAIT $pdo, $id)
```

#### 8.7.23.2 Description

Cr  e l'objet correspondant   un identifiant syst  me

#### 8.7.23.3 Param  tres

PDOAIT **\$pdo**                      pointeur sur la base de donn  es

integer **\$id**                      identifiant syst  me

#### 8.7.23.4 Retour

Retourne NULL ou une instance d'objet (AIT\_ItemType, AIT\_Item, AIT\_TagType, AIT\_Tag)

#### 8.7.23.5 Exemple

```
<?php
require_once 'prepend.php';

$o = AIT::getBySystemID($db, 17);
echo $o->get() . " : ";

$o = AIT::getBySystemID($db, 18);
echo $o->get() . ", ";

$o = AIT::getBySystemID($db, 19);
echo $o->get() . "\n";

// Affichera : a1902425 : Heroes, Antonio Vivaldi
```

## 8.8 AIT

Cette classe poss  de quelques m  thodes statiques

## Chapter 9

# Maintenance

Voici quelques requêtes SQL permettant de contrôler l'état d'une base AIT :

### 9.1 Contrôler la cohérence de la base

Voici quelques requêtes pour vérifier que des lignes les jointures entre tag soit toujours vrai.

```
SELECT tagged.* FROM tagged LEFT JOIN tag ON tagged.item_id=tag.id WHERE tag.id IS NULL
SELECT tagged.* FROM tagged LEFT JOIN tag ON tagged.tag_id=tag.id WHERE tag.id IS NULL
SELECT a.* FROM tag a LEFT JOIN tag b ON a.type=b.id WHERE a.id <> 1 and a.id <> 2 and b.id ←
IS NULL
```

### 9.2 Visualiser les tags non utilisés

C'est à dire les tags non associés à un item.

```
SELECT *
FROM tag a LEFT JOIN tag b ON a.type=b.id
WHERE b.type=2 and a.frequency=0
```

```
SELECT *
FROM tag a LEFT JOIN tag b ON a.type=b.id LEFT JOIN tagged c ON a.id=c.tag_id
WHERE b.type=2 and a.frequency=0 and item_id IS NULL
```

```
SELECT *
FROM tag a LEFT JOIN tag b ON a.type=b.id LEFT JOIN tagged c ON a.id=c.tag_id
WHERE b.type=2 and item_id IS NULL
```

### 9.3 Visualiser les items non utilisés

C'est à dire les items ne possédant aucun tag.

```
SELECT *
FROM tag a LEFT JOIN tag b ON a.type=b.id LEFT JOIN tagged c ON a.id=c.item_id
WHERE b.type=1 and c.tag_id IS NULL
```

§§§